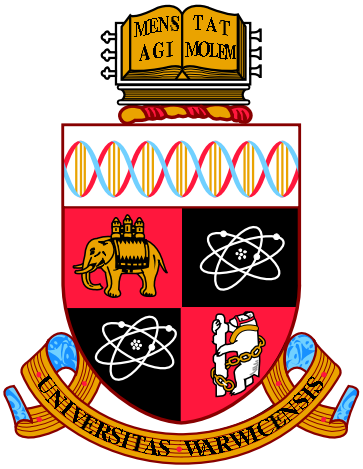


Reconstructing regulatory networks from high-throughput post-genomic data using MCMC methods

by

Sapna Sharma

Thesis

Submitted to The University of Warwick

for the degree of

Doctor of Philosophy

Systems Biology Centre

April 2013

THE UNIVERSITY OF WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I sincerely thank my advisor Prof David Wild for his guidance throughout this thesis and Dr Francesco Falciani for his guidance in biological part of the thesis. Regular meetings with full of valuable advice had provided immense energy and motivation throughout this work. I would like to thank other members of our group for sharing knowledge and for proof reading, includes Chris Penfold, Rich Savage, Emma Cooke and Paul Kirk. I should also like to thank Paul Brown for his useful advice on use of the WSB cluster and CSC cluster. Other members include Katherine Denby for useful lectures and microarray experiments based on advance bioinformatics, Siren Veflingstad for proof reading.

I would like to thank David Hodgson and Hugo Van der Berg for helpful discussions. Amongst many others I would like to thank Anna Stincone from University of Birmingham for sharing *E. coli* data for acid stress condition. I am thankful for constructive critics from my panel committee, Andrew Stuart, Gareth Roberts and Fabio Rigat. In the end I would like to thank entire staff of the Systems Biology Centre for sharing quality of time.

I should thank all my dear friends from UK and India for their support during this PhD, in particularly John Rogers and Katia Merine.

Last but not least I would like to thank my lovely parents, brother Sandeep and my husband Franz for being there for me with constant love, encouragement and patience. Warm tribute to late family members whom I lost last year.

Declarations

I declare that this thesis lies within the University of Warwick regulation and represents my own piece of work except for where otherwise noted.

Signed: Sapna Sharma

Date: 22^{nd} July 2011

Abstract

Modern biological research aims to understand when genes are expressed and how certain genes influence the expression of other genes. For organizing and visualizing gene expression activity gene regulatory networks are used. The architecture of these networks holds great importance, as they enable us to identify inconsistencies between hypotheses and observations, and to predict the behavior of biological processes in yet untested conditions.

Data from gene expression measurements are used to construct gene regulatory networks. Along with the advance of high-throughput technologies for measuring gene expression statistical methods to predict regulatory networks have also been evolving. This thesis presents a computational framework based on a Bayesian modeling technique using state space models (SSM) for the inference of gene regulatory networks from time-series measurements.

A linear SSM consists of observation and hidden state equations. The hidden variables can unfold effects that cannot be directly measured in an experiment, such as missing gene expression. We have used a Bayesian MCMC approach based on Gibbs sampling for the inference of parameters. However the task of determining the dimension of the hidden state space variables remains crucial for the accuracy of network inference. For this we have used the Bayesian evidence (or marginal likelihood) as a yardstick. In addition, the Bayesian approach also provides the possibility of incorporating prior information, based on literature knowledge.

We compare marginal likelihoods calculated from the Gibbs sampler output to the lower bound calculated by a variational approximation. Before using the algorithm for the analysis of real biological experimental datasets we perform validation tests using numerical experiments based on simulated time series datasets generated by *in-silico* networks. The robustness of our algorithm can be measured by its ability to recapture the input data and generating networks using the inferred parameters.

Our developed algorithm, GBSSM, was used to infer a gene network using *E. coli* data sets from the different stress conditions of temperature shift and acid stress. The resulting model for the gene expression response under temperature shift captures the effects of global transcription factors, such as *fnr* that control the regulation of hundreds of other genes. Interestingly, we also observe the stress-inducible

membrane protein OsmC regulating transcriptional activity involved in the adaptation mechanism under both temperature shift and acid stress conditions. In the case of acid stress, integration of metabolomic and transcriptome data suggests that the observed rapid decrease in the concentration of *glycine betaine* is the result of the activation of osmoregulators which may play a key role in acid stress adaptation.

Chapter 1

Introduction

1.1 Biological concepts

In any living system, the cell is the basic functional and structural unit. There are two types of cell, namely eukaryotic and prokaryotic. Prokaryotes are unicellular organisms without a nucleus. Eukaryotic cells accommodate complex structures caged within membranes, and have a well defined nucleus. Within a cell, a double helix deoxyribonucleic acid (DNA) molecule contains the genetic instruction.

DNA is a long stretch of linked nucleotides that consists of adenine (A), cytosine (C), guanine (G) or thymine (T). The DNA is composed of a double helix (strands of nucleotides). The two single strands are arranged antiparallel, and are linked with hydrogen bonding between complementary bases (A pairs with T and C with G). As the strands are complementary, it is sufficient to represent a DNA molecule by a single strand, with one end called $5^{'}$ and other end called $3^{'}$ [Alberts et al., 2002]. DNA sequences are conventionally written in $5^{'}$ to $3^{'}$ direction. These numbers are the index of carbon atoms on the deoxyribose. Deoxyribose is an essential sugar forming the backbone of a DNA molecule.

Amino acids are the building blocks of proteins. A gene is a part of a DNA molecule with a specific sequence that carries the information required to construct

a particular protein. In eukaryotes, the parts of a gene that code for protein are known as exons. Each exon is separated by non-coding regions known as introns. A chromosome is a large portion of DNA that has many genes and non-coding protein sequences. Different organisms have a different number of chromosomes. As an example, a unicellular bacterium like *E. coli* has a single chromosome, whereas higher organisms, for example humans, have two sets of 23 chromosomes, although the complexity of an organism does not depends on its number of chromosomes.

Genes are transcribed into messenger ribonucleic acid (mRNA) through a process called transcription. mRNA is translated into protein with the help of cellular machinery called the ribosome. In some cases, when mRNA may not be translated into protein, this will result in functional RNA (fRNA) also known as non-coding RNA. Transcribed non-coding RNA is called an RNA gene (e.g., ribosomal RNA (rRNA), transfer RNA (tRNA)). As summarised in Figure 1.1, the process of transcription of DNA to RNA and the translation of RNA to proteins is referred to as the central dogma of molecular biology [Crick, 1970].

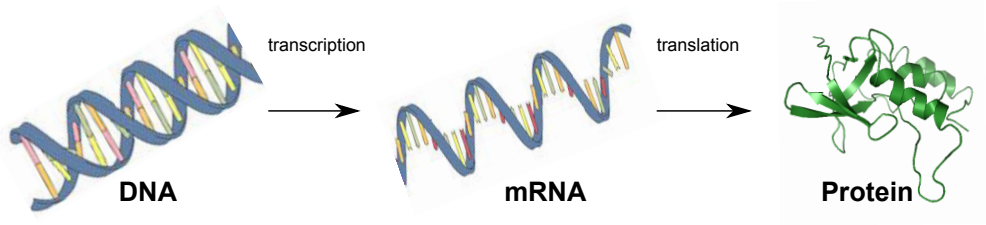


Figure 1.1: Represents the basic transcription and translation activity within the cell. Transcription is the synthesis of mRNA from DNA. mRNA contains genetic information copied from a portion of DNA. mRNA further translates into protein.

The generation of mRNA molecules provides the biological information contained within a gene and this assurance allows a gene to be expressed. Generally, all genes are present in a cell but, depending on the function of the cell, only a fraction of genes are used to produce mRNA at any given time. As we know, the cell lives in a very complex environment, and so faces different types of signals from

internal and external sources. It responds to different environments by producing appropriate proteins that can act in response. In order to identify the genes that are differentially expressed between two conditions researchers can use gene expression profiling. A detailed description of ways to measure expression profiles are described in Section 1.1.1. Since the case studies presented in this thesis deal with prokaryotic cells, the following describes a few of the basic principles of gene regulation in a prokaryotic cell [Alberts et al., 2002].

Basic principles of gene regulation in prokaryotes

In a prokaryotic cell the process of transcription and translation occurs in parallel. This is due to the fact that in a prokaryotic cell the genetic material is not enclosed in a nucleus and therefore gets access to ribosomes in the cytoplasm. Transcription can be controlled by a variety of regulators, also known as transcription factors. The transcription process needs to be initialised. For this the enzyme RNA polymerase (RNAP), a DNA-binding protein, binds to a specific (sigma) DNA binding site. The promoter is located upstream of the genes. Binding of RNAP to the promoter with the help of catabolite activator protein (CAP) starts the transcription process. Promoters vary in strength, meaning how tightly RNA polymerase with its associated proteins binds to the promoter region on the DNA. Transcriptional regulation comes from transcription factors, which can influence the stability of the CAP at initiation. For the termination of a transcriptional process there are two mechanisms, intrinsic termination and rho-dependent termination. Intrinsic termination is also known as rho-independent termination and involves terminator sequences within the RNA that controls RNAP and stops the process. However the rho-dependent mechanism uses the sigma (σ) factor protein to stop RNA synthesis by binding at a specific rho utilisation site.

In E. coli there are two well-studied positive and negative gene regulation systems, known as the *lac* operon and *trp* operon respectively. In such operons bacterial genes involved in related functions are located adjacent to each other and

can be regulated co-ordinately. Therefore, in presence of a suitable inducer, the set of genes can be expressed. For example, the *lac* operon consists of the *lacZ*, *lacY* and *lacA* genes and their transcription is regulated in the presence of lactose. This is an example of positive regulators as in the presence of lactose the set of *lac* operon genes expressesd and encode β-galactosidase, lactose permease and thiogalactoside transacetylase. The transcription of tryptophan genes is regulated by the presence or absence of a co-repressor called tryptophan. This is an example of negative regulation as the presence of tryptophan prevents the expression of the *trp* genes and in the absence of tryptophan the *trp* genes express.

1.1.1 Measuring gene expression

Why measure gene expression? Any observed change in the gene expression highlights an event. A gene expression profile provides a snapshot of transcriptional activity at the molecular level. It can also represent the collective interactions of many events or phenomena that are difficult to detect. In short, it can be regarded as a proxy for a transcriptional event.

Gene expression profiling with microarray technology has become a standard procedure to view the response of an organism or cell under a single or many treatments. Details about microarray techniques are given in Section 1.1.2. Recently, next-generation DNA sequencing technologies (NGS), have emerged as an alternative method for sampling the transcriptome [Kwon and Ricke, 2011]. Microarrays identify gene expression by hybridization and quantification of probes using florescence intensity, whereas in NGS technology, the identification of gene expression can be undertaken by sequencing DNA and quantifying transcripts through the count of the number of sequences that align to a reference transcript.

1.1.2 DNA Microarrays

In the past twenty years, there have been remarkable developments in the field of DNA microarray technology. Microarray devices enable us to measure the expression of thousands of genes in parallel and have revolutionised the field of biological science. The principal feature of this technology is the volume of quantitative data that it can generate. This provides an opportunity for complex molecular processes to be investigated. In this thesis we will be using data that offers the possibility to reverse engineer a model of the transcriptional control system of a bacterial cell's responses to stress.

A microarray is a slide of glass that has a high density array that contains thousands of features defined by fragments of DNA (known as probes) fixed on the glass surface. These probes indicate the presence of their complementary sequence in the target sample. This can be done by base pairing of the probe and the target mRNA (See Figure 1.2). A higher number of complementary base pairs in a pair of nucleotide sequences results in tighter non-covalent bonding between two strands. The unbound sequences are cleared from the glass surface in such a way that only strongly paired strands remain hybridized. Probe sequences bonded with fluorescently labelled target sequences generate a signal that depends on the hybridization conditions such as temperature. The fluorescent signal on the slide is from Watson−Crick base pairing. From each of the samples the microarray measures the level of fluorescent signal [Stekel, 2003].

In this section, we will briefly review different ways of constructing DNA microarrays. Basically, DNA microarrays fall into two categories; (1) those that can be constructed in a lab and (2) those that are produced by commercial companies. In 1994, the first cDNA array was developed by Pat Brown's lab at Stanford University [Mark, 2000, DeRisi et al., 1997]. This was the first so called "*home brew*" or "*roll your own*" glass slide microarray, which was produced in a home lab environment. Mark [2000] introduced the high speed robotic printing of cDNA on glass. The

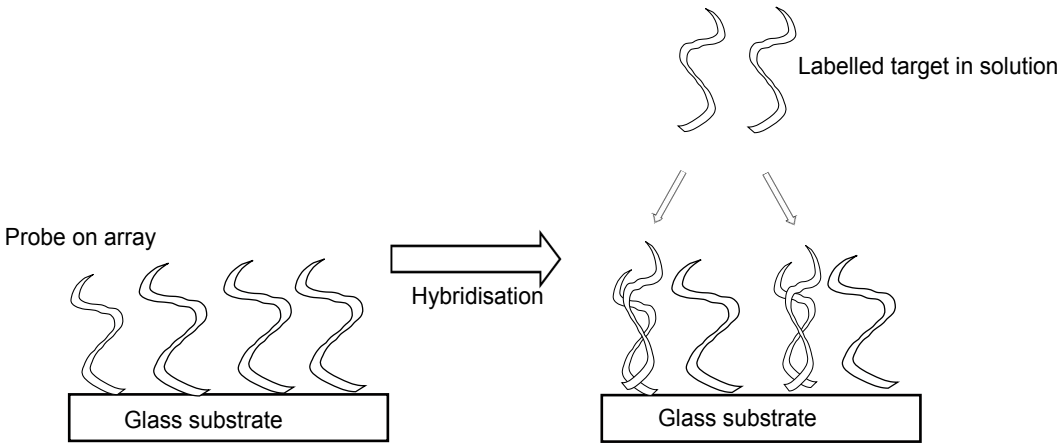


Figure 1.2: Hybridisation is a process by which labelled targets in solution form heteroduplexes with probes on the array through base pairing between the probes and the targets.

second format is the manufactured array, of which the best known is the Affymetrix GeneChip format. In addition to these two well-known array formats there are other formats offered by commercial companies such as Agilent, Nimblegen, Oxford Gene Tech, Xeotron, Combimatrix, Febit and Nanogen. Each of these formats is more or less related in concept to the spotted array or Affymetrix format [Irizarry et al., 2003]. Detailed descriptions of each of these designs are described in Falciani [2007, Chapter 2].

In addition to transcriptional profiling, we are also interested in studying metabolic profiling. Metabolomics is the study of chemical processes involving metabolites. Metabolites are small molecules that are intermediates in or are end products of metabolism. Therefore, in cases where transcription profiling fails to unveil the complete picture of events that take place in a cell, metabolite profiling can fill the gaps with an immediate snapshot of the cell physiology.

Figure 1.3 from [Brazhnik et al., 2002],[Penfold and Wild, 2011] provides a schematic example that explains the relation between genes, proteins and metabolites. A schematic diagram of gene network consists of four genes, and includes a

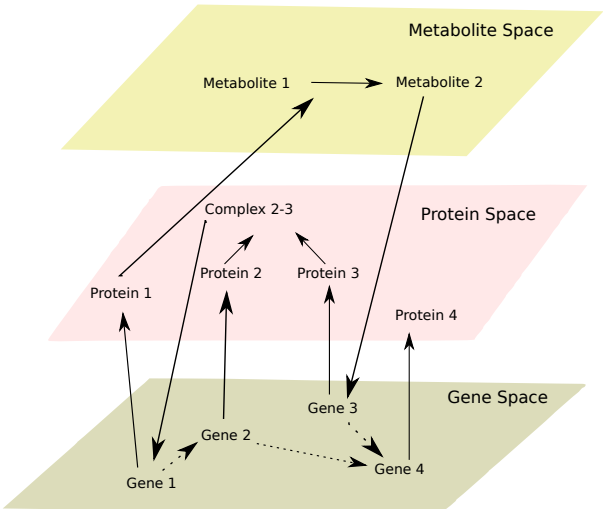


Figure 1.3: This diagram of a gene network is adapted from [Brazhnik et al., 2002], where nodes are organized in gene, protein and metabolite spaces. In this network solid arrows simply indicate the interactions without the signs of activation or repression. Two different mechanisms of gene-gene interactions can be observed here: (a) gene 1 is regulated by the complex 2-3 which is formed by the products of gene 2 and gene 3; (b) gene 3 is regulated by the metabolite 2 which is produced by the protein product of gene 1.

transcription factor *complex* $2-3$ that regulates gene 1. The protein product of gene 1 acts as a catalyst that triggers the production of metabolite 2 from metabolite 1. In practice, the task of integrating transcriptomic and metabolomic information can be challenging, though if possible, it can provide a more complete picture of the organism studied.

In the following section, we will start with a brief introduction to the process of measuring metabolic profiles. Further investigation of metabolic profiling is presented in Chapter 6.

1.1.3 Nuclear magnetic resonance

Nuclear magnetic resonance (*NMR*) is concerned with the magnetic properties of certain atomic nuclei. It can offer an impressive amount of information about a molecule, such as its structural properties, dynamics, reaction state and chemical

environment. Metabolic information can typically be extracted by observing the one-dimensional ($1D$) ^1H nuclear magnetic spectrum. ^1H *NMR* is also known as proton *NMR*. By providing additional electromagnetic radiation it is possible to excite hydrogen nuclei to a higher energy level. The energy required for nuclei to excite depends on the strength of the external magnetic field used; it is usually in the frequency range of about $60 - 100MHz$. When conducting a measurement, the NMR instrument generates a spectrum (for an example see Figure 1.4). In such a spectrum, it is possible to observe the interaction between the resonant frequency and the proton as it is excited from one orientation to an other as a peak. The excitation of a proton from one magnetic alignment to another through a certain frequency is known as the resonance condition. [Keeler, 2007, Chapter 2]

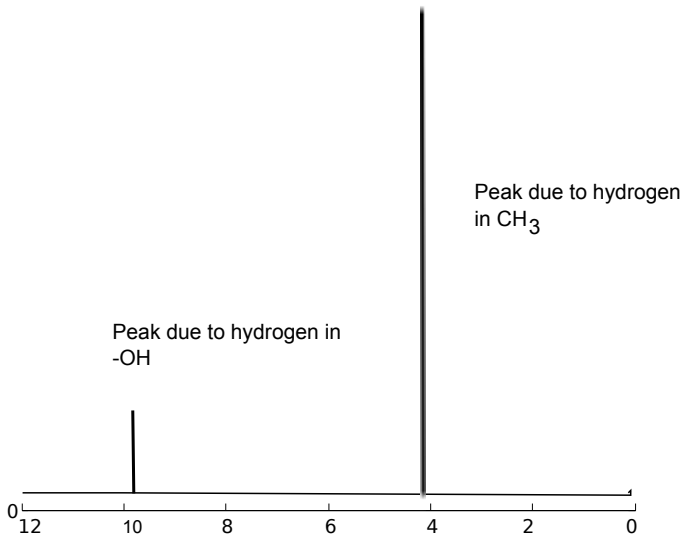


Figure 1.4: An example of the simple *NMR* spectrum of *ethanoic* acid CH_3COOH.

As an example for *ethanoic* acid, the *NMR* spectrum shows two peaks, because of two different environments for hydrogen in the CH_3 group and the $COOH$ group. The peaks are at different places based on the requirement of different external magnetic fields to get resonance of the ^1H nuclei (also known as chemical shift). The peak size conveys the number of H atoms in each group.

So far we have covered the basics of *NMR* spectroscopy. The advanced application of *NMR* spectroscopy is well studied for the stress responses of different organisms by [Viant, 2003], [Viant et al., 2003], [Gavaghan et al., 2011] and [Maher et al., 2012]. However in the Chapter 6 of this thesis we focus on the pre and post analysis of an *NMR* metabolite dataset based on bacterial cells.

1.2 Statistical concepts and methods

In this section we describe some of the well known statistical principles and techniques that will be used in the following chapters of this thesis. Depending on the nature of a given problem, statistical analysis defines probabilistic assumptions about the data by introducing a parametrised probabilistic model (by assigning a probability distribution). These parameters are initially unknown, and therefore need to be inferred, in order to explain the data in the best possible way and also to make predictions. Inference of parameters can be achieved by using a Classical or Bayesian approach [Gilks et al., 1996].

1.2.1 Classical analysis

The key concept that fits the classical approach in statistical modelling is the use of procedures inspired by the "classical" objective of Hypothesis Testing and Parameter Estimation. Hypothesis tests are based on acceptance or rejection of the null hypothesis that the data are assumed to follow. For example, to confirm whether a set of independent and normally identically distributed (*iid*) random variables $X_1, \ldots, X_n$ with unidentified variance have a mean μ. Based on the sample mean $\bar{X}$ the distribution of $\bar{X}$ may be approximated, for example by using a Student's t-distribution, also known as t-statistic. The t-statistic can be applied to explain the difference between the assumption and the data. From this a p-value is calculated that gives the probability that the variables can take more extreme values than the

ones actually observed, assuming that the initial hypothesis holds.

Alternatively, when we want to compare the distribution of two samples one can also apply a non-parametric approach. These are free from assumptions that the data are drawn from a given probability distribution, such as a permutation test. This test evaluates whether the difference in mean values between the samples is significant or not. In a two sample permutation test we randomly re-label the observations from each group drawn from the two samples and estimate an empirical distribution based on the difference of their means.

In the case of the estimation of parameters, the likelihood for the parameter $\mathbf{Y}$, $L(\boldsymbol{\theta}) = f(\mathbf{Y}, \boldsymbol{\theta})$ is considered as a function of the parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ is the parameter space. Parameter estimation is based on finding the parameter value that maximises likelihood. This is also known as the Maximum Likelihood estimation method defined as $\hat{\boldsymbol{\theta}}_{ML} = argmax(L(\boldsymbol{\theta}|\mathbf{Y}))$ [Hogg et al., 2012].

1.2.2 Bayesian analysis

Within a Bayesian framework, the parameters of the model $\boldsymbol{\theta}$ are treated as random variables (with some defined probability distribution). Before any data is observed, a prior distribution can be used to express prior beliefs about the parameters. The Bayesian approach is subjective, as it incorporates personal belief about the distribution of parameters. However, there are non-informative priors, meaning that in the absence of any prior information, one can adopt a flat prior across the range of possible values of theta. A flat prior reflects ignorance about parametric knowledge. Often less informative priors are preferred, having a minimal influence on the posterior distribution [Gamerman and Lopes, 2006].

The data $\boldsymbol{Y}$ can be modelled based on the parameters $\boldsymbol{\theta}$. The $\boldsymbol{\theta}'$s are random quantities with prior probability distribution $P(\boldsymbol{\theta})$. According to Bayes' theorem,

the posterior distribution can be defined for the parameters, $\boldsymbol{\theta}'$s, given the data $\boldsymbol{Y}$,

$$P(\boldsymbol{\theta}|\boldsymbol{Y}) = \frac{f(\boldsymbol{Y}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{\int_{\Theta} P(\boldsymbol{Y}|\boldsymbol{\theta})P(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{1.1}$$

In the absence of knowledge of the denominator of equation 1.1, the posterior is approximated as

$$P(\boldsymbol{\theta}|\boldsymbol{Y}) \propto f(\boldsymbol{Y}|\boldsymbol{\theta})P(\boldsymbol{\theta}). \tag{1.2}$$

Often, calculation of the posterior distribution $P(\boldsymbol{\theta}|\boldsymbol{Y})$ requires an evaluation of higher dimensional integrals which are numerically intractable. In order to deal with such complexity, we need to employ approximation techniques, which can be implemented using Markov Chain Monte Carlo (MCMC) methodologies. MCMC algorithms simulate a random variable, $\boldsymbol{x}$, such that the sequence $x_1, x_2, \ldots$ forms a Markov chain with a specified equilibrium distribution. In a Bayesian context this equilibrium distribution is the posterior distribution. If new point x_{n+1} depends only on the previous point x_n then the chain possesses the Markov property. The chain i.e., collection of simulated samples from posterior distribution will then be used to draw conclusions concerning parameter estimation (or model prediction) based on statistical measures such as mean and variance or other measures calculated from the samples [Carter and Kohn, 1996].

Metropolis-Hastings

The insight behind the Metropolis-Hastings algorithm is the notion of a reversible chain. A Markov chain is said to be reversible if the probability of a state x, $\pi(x)$, with transition probability $T(x^{'}|x)$ is such that

$$T(x^{'}|x)\pi(x) = T(x|x^{'})\pi(x^{'}). \tag{1.3}$$

This condition is also known as detailed balance based on the fact that a Markov

chain is said to be detailed balance if and only if it is a reversible Markov chain. The equation 1.3 is 'balanced' due to the symmetric roles of states x and $x^{'}$. It is called 'detailed' as it holds for every possible pair of states.

Assume a sequence of random variables $X_1, X_2, \ldots, X_t$, generating a sample from the target density f as $x_1, x_2, \ldots, x_t$. The basic idea of the Metropolis-Hastings sampling is to generate a Markov chain that has the target density f as its equilibrium density. To do so the Metropolis-Hastings algorithm is set as below:

Step 1 Sample a candidate value x^* for X_{t+1} from the proposal density $Q(x_{t+1}|x_t)$.

Step 2 Given the candidate value x^*, calculate the acceptance probability $\alpha(x^*|x_t)$ as:

$$\alpha(x^*|x_t) = min\left(\frac{Q(x_t|x^*)f(x^*)}{Q(x^*|x_t)f(x_t)}, 1\right)$$

Step 3 If $\alpha(x^*|x_t) = 1$ then the candidate x^* is accepted and x_{t+1} is set to be x^*. If $\alpha(x^*|x_t) < 1$, then the candidate x^* is accepted with probabilty $\alpha(x^*|x_t)$. The probability of $\alpha(x^*|x_t)$ is set as follows:

- sample randomly a value u from the uniform distribution $U(0, 1)$ based on an interval of $(0, 1)$;
- If $u \leq \alpha(x^*|x_t)$, then candidate value x^* is accepted and set $x_{t+1} = x^*$; otherwise reject x^* and set $x_{t+1} = x_t$.

Repeat steps $1 - 3$ untill a full set of sample $x_1, x_2, \ldots, x_N$ has achieved.

Gibbs Sampling

Gibbs sampling is a special case of MCMC in which proposals are always accepted. Gibbs sampling is for multivariate target densities and simulates a multivariate density using univariate conditional distributions known as the full-conditional distributions. Here, we discuss the simplest Gibbs sampling approach to carry out Bayesian inference [Carter and Kohn, 1996],[Kim and Nelson, 2001]. Suppose for some $k \geq 1$, the k-dimensional multivariate random variable vector $\boldsymbol{\theta}$ can be written

as $\boldsymbol{\theta} = (\theta_1, \cdots \theta_k)$. Suppose the corresponding univariate conditional densities are $f_1, \cdots, f_k$. We assume that we know how to sample from the full conditionals

$$\theta_i|\theta_1, \theta_2, \cdots, \theta_{i-1}, \theta_{i+1}, \cdots, \theta_k \sim f_i(\theta_i|\theta_1, \theta_2, \cdots, \theta_{i-1}, \theta_{i+1}, \cdots, \theta_k)$$

for $i = 1, 2, \cdots, k$.

The associated Gibbs sampling algorithm can be given as a transition from $\boldsymbol{\theta}^{(t)}$ to $\boldsymbol{\theta}^{(t+1)}$. where t is the iteration number,

1. Given starting values $\boldsymbol{\theta}^0 = (\theta_1^0, \cdots, \theta_k^0)$, set $t = 0$;

2. Sample for $t = 1, 2 \cdots, N$

$$\begin{aligned}
\theta_1^{(t+1)} &\sim f_1(\theta_1|\theta_2^{(t)}, \cdots, \theta_k^{(t)}); \\
\theta_2^{(t+1)} &\sim f_2(\theta_2|\theta_1^{(t+1)}, \theta_3^{(t)}, \cdots, \theta_k^{(t)}); \\
&\vdots \\
\theta_k^{(t+1)} &\sim f_k(\theta_k|\theta_1^{(t+1)}, \cdots, \theta_{k-1}^{(t+1)}).
\end{aligned}$$

3. Set $t = t + 1$ and repeat from step 2.

The advantage of a Gibbs sampler is its use of density functions for simulation. Therefore in the case of high-dimensional problems these distributions can also be defined as univariate. Using samples drawn from the full conditional distribution, we can make estimates of the parameters.

In spite of their popularity and wide application, there are several issues that arise in implementing MCMC methods, such as blocking, updating order in Gibbs sampling, defining the optimal number of chains, starting values, determining burn-in, determining stopping time and analysis of the output [Cowles and Carlins, 1996]. Therefore the implementation of such tasks will require fine programming with very careful diagnostic tests to obtain confident results.

1.2.3 The marginal likelihood and model selection

From the Bayesian point of view, model comparison captures uncertainty in the choice of the model. Let us assume that we want to compare a set of L models, i.e. M_i, where $i = 1, \cdots, L$. All of these models define a probability distribution over the observations D. Also assume that the data are generated from one of these models and we do not know which model is the true one. Our uncertainty can be expressed through a prior distribution over the models, i.e. $P(M_i)$. Therefore given a set of data, D, the posterior distribution can be written as

$$P(M_i|D) \propto P(M_i)P(D|M_i).$$

For simplicity we assume that the prior is equally probable among all models. The interesting term to observe here is *the model evidence* which is also known as *marginal likelihood*, $P(D|M_i)$, which shows the preference provided by the data for different models. In other words one can see the *marginal likelihood* as a likelihood function over the space of models, where the parameters have been marginalized. Jeffreys [1961], Kass and Raftery [1995], as well as Berger and Pericchi [2001], proposed the Bayes factor for comparing models M_1 and M_2

$$B_{12} = \frac{P(M_1|D)}{P(M_2|D)} \Big/ \frac{P(M_1)}{P(M_2)}$$

There are other standard frameworks for model selection which we can implement; for instance Schwartz's criterion, which is also called the Bayesian Information Criterion (BIC) [Schwarz, 1978]. The BIC provides a first order approximation of the Bayes factor, and requires the maximum likelihood estimation (MLE) of parameters for all models.

$$S = -2 \log \lambda_n - (p_2 - p_1) log(n)$$

where $\lambda_n = L_{1,n}/L_{2,n}$ is the log-likelihood ratio for the comparison of models M_1

and M_2 evaluated at the MLE, $p_1,\ p_2$ are the dimensions of the parameter space associated with M_1 and M_2 and n is the sample size.

Based on *deviance*, Spiegelhalter et al. [2002] developed an alternative to the BIC, called the DIC (for Deviance Information Criterion). For Bayesian model selection or comparision DIC is particulary prefered. The deviance is defined as $D(\boldsymbol{\theta}) = -2\log(P(\boldsymbol{Y}|\boldsymbol{\theta})) + C$, where $\boldsymbol{Y}$ is the data, $\boldsymbol{\theta}$ is unknown parameter and $P(\boldsymbol{Y}|\boldsymbol{\theta})$ is the likelihood function. The constant C will be cancel out on comparision of different models. Based on the deviance of the model, the deviance information criterion (DIC) can be calculated as

$$DIC = D[E(\boldsymbol{\theta})] + p_D,$$

where $E(\boldsymbol{\theta})$ is the expectation of $\boldsymbol{\theta}$ and p_D computes the effective number of parameters,

$$p_D = E[D(\boldsymbol{\theta})] - D[E(\boldsymbol{\theta})],$$

where $E[D(\boldsymbol{\theta})]$ is the posterior mean of the deviance term, that measures the strength of the model fitting the data. The DIC is then calculated for the evaluation of the model. Providing the DIC value is smaller, the model is regarded as better. This criterion is more satisfactory when compared to BIC. Firstly, because it considers the prior information and gives a natural penalization factor to the log-likelihood; secondly, because the DIC can easily be calculated from MCMC simulated samples.

Finally, we describe Bayesian evidence (or marginal likelihood) as a yardstick for model selection. The obvious question to raise here is "why use a marginal likelihood for model selection?". This can be answered by considering the principle of "Ockham's Razor". This principle states a preference for simple models. Bayes' theorem may be used to rank models by comparing how well they predict the data. These predictions are based on model evidence. As shown in Figure 1.5 a simple

model makes only a certain range of predictions for the data, whilst a more complex model will be freer to predict multiple datasets. This means that a simple model may still predict the data more strongly than a complex one and hence fulfills the Ockham's Razor principle.

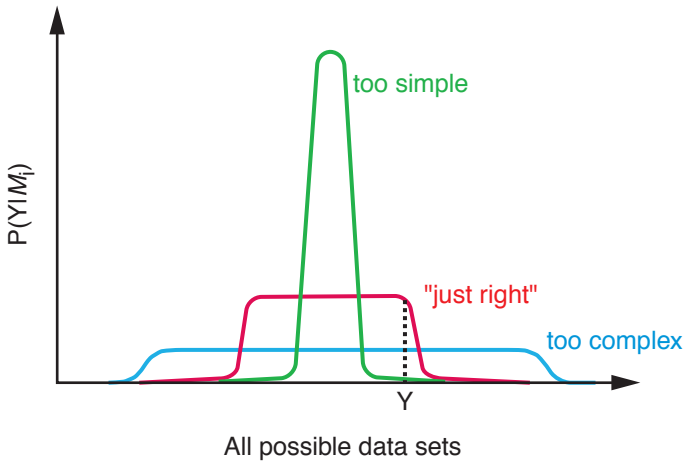


Figure 1.5: Model classes may be either too simple or too complex to generate the data set. In such cases computing marginal likelihood gives a probabilistic yardstick for selection of the model class [MacKay, 2003].

In the practice of Bayesian statistics, the use of MCMC methods to simulate the posterior distribution is widespread [Gelfand and Smith, 1990]. Once sufficient samples have been drawn from the posterior distribution, one can tackle or solve the problem of estimation and prediction very well by using these methods. However, calculation of the model evidence has proved extremely challenging. Chib [1995] demonstrates a method to compute marginal likelihoods using Gibbs sampler output. Chib's method gives the simplest way to compute the marginal likelihood, given parameters drawn from the posterior distribution.

We will later compare marginal likelihood calculations for State Space Models using the outputs of the Gibbs sampler [Chib, 1995], to the lower bound calculated by the variational approximation [Beal et al., 2005]. The variational approximation has its roots in the 'calculus of variations'. Recently, variational methods have been used in the context of approximate inference and estimation. Using the variational

free energy as a framework for statistical inference, an ensemble of parameter vectors is optimised, rather than a single parameter vector [MacKay, 1995]. This method was utilised by Beal et al. [2005] in the reconstruction of genetic regulatory networks using hidden factors.

1.2.4 Introduction to Gaussian processes

A Gaussian process is a stochastic process where any finite set of random samples has a multivariate normal distribution [Rasmussen and Williams, 2006]. Gaussian Process Regression (GPR) is a non-linear regression method and has been widely used for time series modelling as well as for gene expression analysis [Stegle et al., 2010], [Kalaitzis and Lawrence, 2011], [Chu et al., 2005].

In GPR we assume, $\boldsymbol{y} = f(\boldsymbol{x}) + \epsilon$ can represent the empirical observations, where $f(\boldsymbol{x})$ represents the latent (or unobserved) gene expression, i.e., the observation is a noisy version of the same underlying true gene expression. We assume that the unobserved function $\boldsymbol{f} : \boldsymbol{x} \rightarrow \Re$, is drawn from an finite dimensional Gaussian distribution, where the correlation function between the points is determined by a covariance function, Σ. We assume ϵ is an *iid* additive noise and follows a normal distribution $N(\epsilon|0, \sigma_\epsilon^2)$ of 0 mean and variance σ_ϵ^2 [Rasmussen and Williams, 2006].

Consider a time series dataset $\boldsymbol{y} = [y_{1,T}, \ldots, y_{G,T}]$ to be of dimension $N = G \times T$ where G represents the total number of genes and T are $[1, \ldots, T]$ timepoints. In order to make inference about a latent function $\boldsymbol{f}$ we will be required to define a prior belief. The use of the Gaussian process as a prior on $\boldsymbol{f}$ justifies the term "Gaussian process model". The Gaussian prior is considered as a non-parametric form, mainly because instead of defining/giving a particular parametric form to $\boldsymbol{f}(\boldsymbol{x})$ the prior is placed on the function value directly; i.e, each element of $x \in \boldsymbol{x}$ is a random variable $\boldsymbol{f}(x)$.

The marginal likelihood is the integral of the likelihood times the prior:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \int P(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{x})P(\boldsymbol{f}|\boldsymbol{x})d\boldsymbol{f}, \tag{1.4}$$

Equation 1.4 shows the marginalization over the function values $\boldsymbol{f}$. In the Gaussian process model the prior on $\boldsymbol{f}$ is Gaussian, $P(\boldsymbol{f}) = N(0, \mathbf{K})$, where $\mathbf{K}$ is the covariance matrix or kernel function. The log prior can be given as;

$$\log P(\boldsymbol{f}|\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{f}^T\mathbf{K}^{-1}\boldsymbol{f} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log(2\pi) \tag{1.5}$$

and the likelihood is a factorized Gaussian $P(\boldsymbol{y}|\boldsymbol{f}) = N(\boldsymbol{f}, \sigma_\epsilon^2 I)$, where I is an identity matrix. Subsituting the log prior and the likelihood in equation 1.4 the integration yields the log marginal likelihood as;

$$\log P(\boldsymbol{y}|\boldsymbol{x}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\boldsymbol{y}^T(\mathbf{K})^{-1}\boldsymbol{y}. \tag{1.6}$$

The same result can also be obtained by noting that $\boldsymbol{y} \sim N(0, \mathbf{K} + \sigma_\epsilon^2\mathbf{I})$. There are various ways to define this covariance function [Rasmussen and Williams, 2006].

1.2.4.1 Covariance function

The covariance function K reflects the relation between the values of the function, f, for a given time point. The covariance function must be positive semidefinite to be valid, i.e. any positive semidefinite $n \times n$ matrix K which satisfies, $r^T K r \geq 0$, for all $r \in R^n$. A symmetric matrix is positive semidefinite iff all its eigenvalues are positive. This indicates that if $r^T K r = 0$ only when $r = 0$ then K is positive definite [Cooke et al., 2011].

A commonly used example is the squared exponential covariance function

which can be parameterised in terms of its hyperparameters as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp(\frac{-1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T M(\boldsymbol{x}_i - \boldsymbol{x}_j)) + \sigma_\epsilon^2 \delta_{ij}.$$

where $\theta = (\sigma_f^2, \sigma_\epsilon^2, \{\mathbf{M}\})$ is a vector of all hyperparameters. Here δ_{ij} is the Kronecker delta function, σ_f^2 is the signal variance, σ_ϵ^2 is the noise variance and $\{\mathbf{M}\}$ denotes the parameters of a symmetric matrix $\mathbf{M}$ with two choices of $\mathbf{M}$ being $\mathbf{M}_1 = \boldsymbol{l}^{-1}I$, $\mathbf{M}_2 = diag(\boldsymbol{l})^{-2}$. Here $\boldsymbol{l}$ is a positive valued vector. The properties of functions using these covariance functions depends on the value of the hyperparameters. The covariance function can be used to access the importance of hyperparameter, whist trying to understand data. However in the case of a squared exponential using the $\mathbf{M}_2$ distance measure, the $l_1 \ldots, l_d$ hyperparameters play the key role of a characteristic length scale; generally l_i defines the length on a particular axis of input space for the function value to be uncorrelated. Such a covariance function can be used to implement automatic relevance determination (ARD) [Neal, 1996]. The inverse of the length scale, l, estimates the relevance of an input; if the length scale is very large, the covariance will almost become independent of this input. In this case, the use of ARD removes such parameters efficiently from inference. Detailed use of ARD is given in Chapter 6[Chu et al., 2005],[Kuss et al., 2005].

1.3 Gene Regulatory Network Inference

The massive acquisition of gene expression profiles can provide a deeper insight into the function of cells. A variety of mathematical formalisms for modelling this type of data have been proposed in the literature. Ventura et al. [2006] provide a wider review of mathematical modelling and its application in biology. So far, these modelling approaches have been most successful for systems of simpler organisms like *E. coli* and *S. cerevisiae* [Cantone et al., 2009].

Given a pre-specified mathematical framework, the behaviour of a group of

genes forming a specific gene regulatory network (GRN) may be simulated under a variety of conditions and used to test hypotheses. Conversely, the observation of gene behaviour under specific conditions may be used to infer the underlying GRN. Generally speaking, the reconstruction of a GRN from the observed measurements is known as a "*reverse engineering*" approach.

In general, there are two well known information extraction approaches, characterised as "top-down" and "bottom-up", which have been applied to inferring GRNs from high-throughput data. A "top-down" approach mainly breaks down a system from experimental observations, in order to gain insights into the system. Alternatively, in a "bottom-up" approach, the researchers attempt to build up a system using observations from different components of the system.

1.3.1 Modelling and reverse engineering approaches

Mathematical and statistical models represent a powerful approach to understand, reflect and describe observations by representing them in terms of a variety of alternative mathematical/statistical frameworks. The benefits of using mathematical models lie in their ability to enhance and augment our understanding of a system, to make quantitative predictions from past and present observations, and to condense previously observed behaviour into a concise framework.

Ordinary differential equation (ODE) models are of a differential equation form that describes the rate of change of gene expression with respect to time, as a function of other gene expression, and as an external perturbation. The model has a differential equation for each of the genes in the network. The parameters of the model are then inferred from the gene expression data.

In information-theoretic approaches, the gene network is reconstructed by considering one pair of genes at a time and checking the co-expression of the two genes across the experimental data set. Evaluation of co-expression between two genes can be done either by correlation or by using a mutual information score

[Bansal et al., 2007].

1.3.1.1 Bayesian Networks

A Bayesian network (BN) describes a directed acyclic graph (DAG) using a probabilistic graphical network model. In the model each node describes a random variable, and edges represent conditional independence relations between random variables. For example an edge from node x to y represents a statistical dependency between variable x and y. Further the arrow indicates that x influences y. Node x is parent of y and y is a child of x. In a broader sense these relations define the set of descendants, the set of nodes that can be reached directly from ancestral nodes. No node can be its own ancestor because of the structure of the acyclic graphs.

A BN reflects the conditional independence statement, such that each variable is independent of its non-descendants in the graph given the state of its parents. This property is very useful to reduce the number of parameters that are needed to define a joint probability distribution of the variables. This reduction also leads to a better estimation of posterior probabilities.

These kinds of relationships are useful to represent gene-gene interactions which can be visualised by a directed graph without cycles. "Without cycles" (acyclic) means a gene may have no direct or indirect interaction with itself. This approach can be used to reverse engineer a gene network by finding the directed acyclic graph that best describes the gene expression data. The particular limitation of a directed acyclic graph can be overcome by using a dynamic Bayesian network if time series observations are available (for more details see next section) [Husmeier et al., 2005].

BNs provide a flexible framework for giving a diagrammatic representation of the probabilistic relationships between sets of variables. In our case, these sets of variables are sets of gene expression measurements, and establishing relationships among these variables will define interactions between the genes. The interactions

between a set of genes can be defined in terms of conditional independence relations [Husmeier et al., 2005]. The overall representation of a BN can be given by a graphical structure $G = (V, E)$ where V are the vertices and E are the edges. G specifies a joint distribution over the set of random variables of interest by defining conditional probability distributions.

For example, consider any given joint distribution $P(x, y, z)$ over three variables x, y and z. By using the product rule of probability the joint distribution can be written as

$$P(x, y, z) = P(z|x, y)P(x, y).$$

By using a second application of the product rule we can factorise $P(x, y)$ as $P(y|x)P(x)$, giving,

$$P(x, y, z) = P(z|x, y)P(y|x)P(x).$$

This factorization is shown graphically in Figure 1.6.

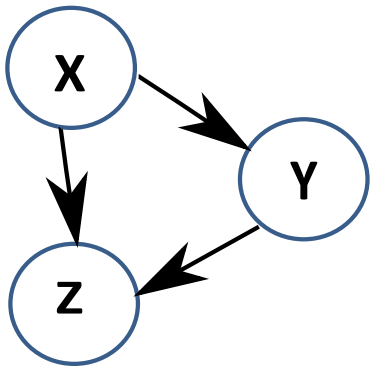


Figure 1.6: A directed graph representing a factorization of the joint probability distribution over three variables x, y, and z.

Despite technological advances in measuring gene expression levels as time series for thousands of genes, the complex nature of the data does not allow us to explore all of the factors that might contribute to genetic regulation and the inter-

actions among genes. Bayesian networks have the advantage of modelling hidden factors, making them very powerful tools for inferring gene networks. However BNs have some limitations, eg. (a) self regulation and feedback loops are likely features in GRNs, but the strict use of a DAG makes it impossible to capture any direct cycle or feedback loops without the use of time series observations and (b) discretization of data for BN analysis may result in a loss of information from continuous gene expression measurements.

1.3.1.2 Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBNs) are Bayesian networks that model sequences of variables. Murphy and Mian [1999] first introduced the use of DBNs to model gene expression data. The benefits of DBNs include the ability to handle latent variables and missing data (such as TF protein concentrations that effect the steady state concentrations of mRNA) and to model stochasticity. Friedman et al. [2000] explored experimental applications to microarray data analysis.

Feedback loops can also be unfolded with respect to time, by explicitly modelling the influence of a gene at time $t = 1$ (i.e. G_1) on another gene at a later time $t = 2$ (i.e. G_2), as shown in Figure 1.7

1.4 State Space Models

We aim to model gene regulatory networks using gene expression time series data and a linear dynamical system (LDS) in a Bayesian framework. The work done by Rangel et al. [2001] shows that biological responses (such as T cell activation) can be effectively modelled. By using such an approach the gene regulatory networks were inferred by fitting LDS models to gene expression profile data collected from microarrays. Later, Wu et al. [2004] described a method to model gene expression, in which a cell can be considered to be a system whose behaviour depends completely

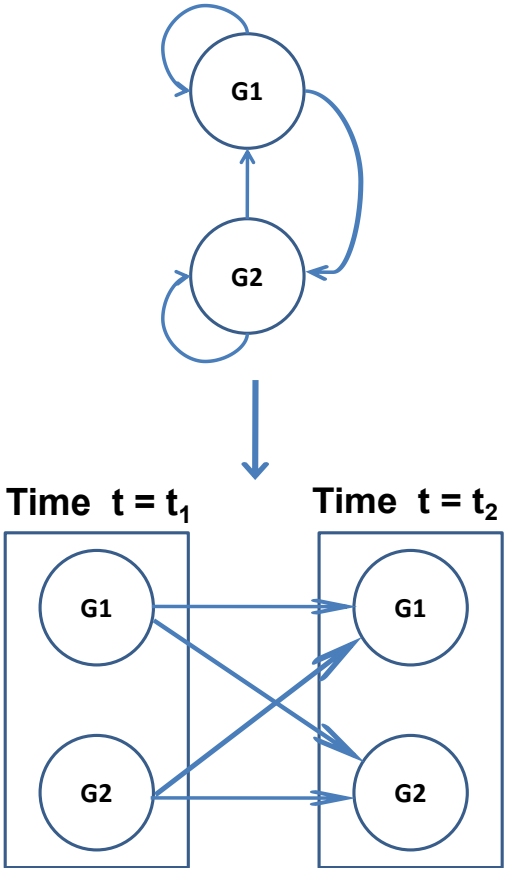


Figure 1.7: Unfolding loops with respect to time steps.

on the current internal state and any external input. The gene expression level in the cell provides information about the response of the cell. The gene expression can thus be modelled using linear state space models. Ong et al. [2002], Perrin et al. [2003], Irizarry et al. [2003] also describe the ability of dynamic Bayesian networks to handle time series data that includes feedback loops and hidden variables.

Linear dynamic systems are state space models (SSM), in which the dynamics of the system can be conveniently and succinctly described by introducing the notions of a state space and state vectors. It is assumed that the system under study can be described by an unobserved sequence of $k-$dimensional real-valued vectors $\{\boldsymbol{x}_t\} = \{x_{k,1}, x_{k,2} \ldots x_{k,T}\}$ which is associated with a series of observations,

and a $p-$dimensional real-valued vector $\{\boldsymbol{y}_t\} = \{y_{p,1}, y_{p,2} \dots y_{p,T}\}$, with respect to development over time ($t = 1, \dots, T$). An SSM is specified by a set of two equations, known as the state and observation equations, and the simplest form of a SSM can be represented as follows:

State equation:

$$\boldsymbol{x}_{t+1} = \mathbf{A}\boldsymbol{x}_t + \boldsymbol{e}_t, \qquad \boldsymbol{e}_t \sim N(0, Q) \tag{1.7}$$

Observation equation:

$$\boldsymbol{y}_t = \mathbf{C}\boldsymbol{x}_t + \boldsymbol{u}_t, \qquad \boldsymbol{u}_t \sim N(0, R). \tag{1.8}$$

Here $\mathbf{A}$ is the $(k \times k)$ state dynamics matrix and $\mathbf{C}$ is the $(p \times k)$ state to observation matrix, $\boldsymbol{e}_t$ and $\boldsymbol{u}_t$ are independent Gaussian noise terms, with error covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ added to the state and observation terms respectively.

If the $\{\boldsymbol{x}_t\}$ and $\{\boldsymbol{y}_t\}$ are both Gaussian distributed, then we obtain a linear dynamical system (LDS) or Gaussian state space model (SSM); these models are therefore also known as linear Gaussian state space models. In our application, the inclusion of hidden variables can model unobserved effects such as the effects of genes that have not been included in the experiment, or the effects of mRNA and protein degradation. Additionally, the ability to handle noisy data makes this type of model very attractive. However, despite model flexibility, there are some issues to be addressed, such as defining an optimal dimension of the hidden state space.

SSMs were originally developed and introduced by control engineers [Kalman, 1960]. The traditional autoregressive model (AR), moving average model (MA) and autoregressive moving average model (ARMA) can be represented in a SSM form in a simple and systematic manner. Conversely, a SSM representation can be put in an ARMA representation. SSMs have the ability to model dynamical systems involving unobserved state variables, making a SSM a special case of a DBN. An extension of the simple state space model to the problem of modelling GRNs is described later.

Aoki [1990] and Rangel et al. [2004] discuss three important properties of SSMs, i.e. stability, observability and controllability. The stability property defines whether the system is asymptotically stable or not. Observability is a measure of how well the internal state of a system can be inferred by knowledge of its external output. The term controllability implies "state control". A system is called controllable if its state variable can directly be controlled by the input. In addition to these properties, it is equally important to check the identifiability property, which a model must satisfy in order for parameter inference to be possible. SSMs are generally unidentifiable, because the hidden state can be rescaled, and accordingly the system matrices of the state and observation equations. This indicates the possibility that two models can give a similar distribution of observations using different values of the hidden variables.

In the past decade, many authors have described research to reverse engineer gene regulatory networks using state space models. For example, the work of Rangel et al. [2001] modelled individual gene interactions using a linear SSM. These authors implemented an Expectation Maximization (EM) algorithm for the estimation of model parameters. Work by Rangel [2003] and Xiong and Choe [2008] describe a method to combine linear dynamical systems modelling with structural constraints, using Lagrange multipliers.

Perrin et al. [2003] proposed a generalized EM algorithm with some constraints on network connections to maximize the model likelihood, but also used a limited choice of hidden state dimensions i.e. $0, 1$, or 2. Wu et al. [2004] used a factor analysis method to identify the internal state variables and the Bayesian information criterion (BIC) to determine the dimensionality of state variables. Beal et al. [2005] use a LDS model to reconstruct regulatory networks from microarray gene expression time series data in a hierarchical Bayesian framework, using a variational Bayesian approach, and calculate a lower bound on the marginal likelihood in order to learn both the network structure and the dimensionality of the hidden

state.

1.5 Thesis outline

Chapter 1 introduces the basic biology and the statistical concepts and methods that support this thesis throughout. Chapter 2 derives a Gibbs sampler algorithm (GBSSM) for a linear dynamical systems, also known as state space model with feedback. This chapter also includes the calculation of model evidence that will be used as a yardstick for the model selection task. In Chapter 3 we demonstrate the validation of proposed algorithm by reverse engineering a toy model using data generated from a state space model. Chapter 4 demonstrates the reconstruction of an *in silico* network and compares the results to those obtained with the variational Bayesian approach.

Chapters 5 and 6 focus on the application of the GBSSM algorithm to infer gene network using high-throughput post-genomic data. In these Chapters we model the adaptation and response of *E. coli* to different stress conditions, such as temperature shift and acid stress. Chapter 6, in addition to transcriptional dynamics we combine metabolomic measurements to understand the underlying biochemical pathways responsible for the adaptation of *E. coli* during acid stress. This thesis concludes with discussion of recent advancements on bacterial studies and suggestions of inferring regulatory networks by using time varying state space models.

Chapter 2

A Gibbs sampler for State Space models

In this chapter we derive and discuss an algorithm based on the Gibbs sampler for State Space Models. To avoid complexity and to understand the idea behind the inference well we have initially shown inference based on the canonical form of a SSM. This is then extended to a SSM with feedback. The extended model also includes replicate information explicitly in the algorithm. For learning hyperparameters we have extended the simple Gibbs sampler to a Metropolis−Hastings within Gibbs algorithm. We also present pseudo−code for the Gibbs sampler that mainly provides computational insight. For the purpose of model selection we also describe the calculation of marginal likelihoods from the Gibbs output.

In summary, this chapter focus on the methodology used to build an MCMC sampling algorithm. In the following Chapter 3, we will show the validation of the proposed algorithm, including the learning of hyperparamters for the SSMs. In Chapter 4 we will demonstrate reverse engineering of an *in silico* network. These two chapters will evaluate the performance of the Gibbs sampler. In subsequent chapters we will study the application of the proposed algorithm using datasets based on biological experiments.

2.1 Model Specification

The state space model as given in equations 1.7 and 1.8 can be extended by utilizing the inputs $\mathbf{y}_{1:T}$ as a feedback:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{y}_{t-1} + \mathbf{w}_t \tag{2.1}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{y}_{t-1} + \mathbf{v}_t \tag{2.2}$$

where $\mathbf{B}(k \times p)$ is the input-to-state matrix and $\mathbf{D}(p \times p)$ is called the input-to-observation matrix. $\mathbf{w}$ and $\mathbf{v}$ are Gaussian noise vectors associated with hidden state and observation respectively. The parameters of the model are summarised in Table 2.1

Parameters	Dimensions	Description
$\mathbf{A} = \{a_{ij}\}$	$k \times k$	captures the state dynamics
$\mathbf{C} = \{c_{ij}\}$	$p \times k$	captures effect of the state on gene level.
$\mathbf{B} = \{b_{ij}\}$	$k \times p$	captures effect of gene level on the state.
$\mathbf{D} = \{d_{ij}\}$	$p \times p$	provides causal gene-gene interaction information.
$\mathbf{w}$ are Gaussian white noise with the diagonal covariance matrices $\mathbf{Q}_{k\times k} = \{q_{ii}\}$		
$\mathbf{v}$ are Gaussian white noise with the diagonal covariance matrices $\mathbf{R}_{p\times p} = \{r_{ii}\}$		

Table 2.1: Description of the parameters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Q}$ and $\mathbf{R}$ in the SSM. Where $a_{i,j}$ represent elements of the matrix $\mathbf{A}$.

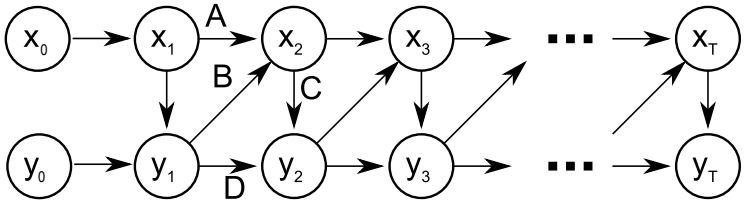


Figure 2.1: The graphical representation of a Gaussian State Space model with feedback following the state and observation equations (2.1 and 2.2) (figure is adapted and modified from Beal et al. [2005]).

In Figure 2.1, $\mathbf{y}_t$ denotes the gene expression levels at time step t and $\mathbf{x}_t$ denotes the unobserved hidden factors of the state space. In practice $\mathbf{y}_t$ is the

vector of suitably normalized values of the gene expression levels. The hidden state of the model represents unobserved quantities such as the expression or degradation level of regulatory proteins or missing gene expression measurements.

The hidden state concentrates on modelling the Markovian dependencies between the successive outputs using the output-input feedback construction. Such models can be a useful for the analysis of gene expression time series data. With this model architecture we aim to discover gene-gene interactions across time steps with the influence of the hidden states. In the following section we show the derivation of a Gibbs sampler considering the cannonical form of a SSM.

2.2 Implementing Gibbs sampling

2.2.1 Canonical State Space Model

The simplest form of a state space model and its graphical representation can be given as

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t \tag{2.3}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t \tag{2.4}$$

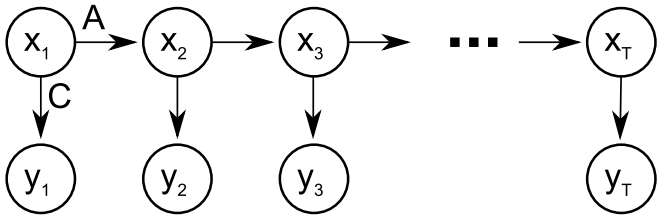


Figure 2.2: Graphical representation of a state space model. Here the hidden state $\mathbf{x}_t$ develops with Markov dynamics as per parameters in $\mathbf{A}$ and at each time step generates an observation $\mathbf{y}_t$ following the parameters in $\mathbf{C}$ (figure is adapted and modified from Beal et al. [2005]).

In the equations 2.3 and 2.4 $\{\mathbf{y}_{1:T}\}$ is a sequence of $p-$dimensional observation vectors. At each time step, t, the observation $\boldsymbol{y}_t$ was generated from a

$k-$dimensional hidden state variable $\boldsymbol{x}_t$. The state $\boldsymbol{x}_t$ at time step t was generated from a $k-$dimensional state variable, $\boldsymbol{x}_{t-1}$, such that the sequence $\{\mathbf{x}_{1:T}\}$, follows a first-order Markov process. The joint probability of a sequence of T states and observation sequences can therefore be expressed as:

$$P(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}) = P(\boldsymbol{x}_1)P(\boldsymbol{y}_1|\boldsymbol{x}_1)\prod_{t=2}^{T} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})P(\boldsymbol{y}_t|\boldsymbol{x}_t) \tag{2.5}$$

The distribution $P(\boldsymbol{x}_1)$ in the equation (2.5) is assumed to be Gaussian, $\boldsymbol{x}_1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\boldsymbol{\mu}_1$ and the $\boldsymbol{\Sigma}_1$ are the initial mean and covariance (generally defined as zero mean and unit covariance),

$$P(\boldsymbol{x}_1) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}_1 - \boldsymbol{\mu}_1)\}}{(2\pi)^{\frac{k}{2}} \det(\boldsymbol{\Sigma}_1)^{\frac{1}{2}}}. \tag{2.6}$$

The state $\{\mathbf{x}\}$ and the observation $\{\mathbf{y}\}$ variables are also assumed to be Gaussian. Following the SSM equations 2.3 and 2.4, the state and the observation variables $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$ can be defined as,

$$\boldsymbol{x}_t|\boldsymbol{x}_{t-1} \sim N(\mathbf{A}\boldsymbol{x}_{t-1}, \mathbf{Q}), \tag{2.7}$$

$$\boldsymbol{y}_t|\boldsymbol{x}_t \sim N(\mathbf{C}\boldsymbol{x}_t, \mathbf{R}). \tag{2.8}$$

In the above equations 2.7 and 2.8, $\mathbf{Q}$ and $\mathbf{R}$ are the $k \times k$ and $p \times p$ dimensional state and observation noise covariances matrices respectively. Therefore, using equations 2.3 and 2.4 we can write the multivariate density functions for the state and observation sequences as:

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1})^T \mathbf{Q}^{-1}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1})\}}{(2\pi)^{\frac{k}{2}} \det(\mathbf{Q})^{\frac{1}{2}}}, \tag{2.9}$$

$$P(\boldsymbol{y}_t|\boldsymbol{x}_t) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{y}_t - \mathbf{C}\boldsymbol{x}_t)^T \mathbf{R}^{-1}(\boldsymbol{y}_t - \mathbf{C}\boldsymbol{x}_t)\}}{(2\pi)^{\frac{p}{2}} \det(\mathbf{R})^{\frac{1}{2}}}. \tag{2.10}$$

For the implementation of the Gibbs sampling algorithm for a simple SSM we will require samples from the complete conditional distribution of the parameters $\boldsymbol{\theta} = (\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R})$ given the state $\{\mathbf{x}\}$ and observation sequences $\{\mathbf{y}\}$ (dropping the subscript $1:T$) i.e. $P(\boldsymbol{\theta} \mid \{\mathbf{x}\}, \{\mathbf{y}\})$. Treating each row of the parameter matrix $\mathbf{A}$ independently, with the i^{th} row of $\mathbf{A}$ denoted by $\boldsymbol{a}_i$ the marginal likelihood $P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{a}_i, \mathbf{Q}_{ii})$ can be given as

$$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = N(\boldsymbol{a}_i \boldsymbol{x}_{t-1}, \mathbf{Q}_{ii}), \tag{2.11}$$

here $\mathbf{Q}_{ii}$ represents the i^{th} element of a diagonal covariance matrix $\mathbf{Q}$ (dimension of $\mathbf{Q}$ is $k \times k$) and it also represents the variance.
By Baye's theorem, the conditional distribution $P(\boldsymbol{a}_i|\mathbf{Q}_{ii}, \boldsymbol{x}_{t-1})$ can be given as

$$P(\boldsymbol{a}_i|\mathbf{Q}_{ii}, \boldsymbol{x}_{t-1}) = \prod_{t=2}^{T} P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) P(\boldsymbol{a}_i). \tag{2.12}$$

The prior $P(\boldsymbol{a}_i)$ is defined as

$$P(\boldsymbol{a}_i) = N(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{\boldsymbol{a}}), \tag{2.13}$$

here $\boldsymbol{\mu}_a$ is $1 \times k$ mean vector and $\boldsymbol{\Sigma}_{\boldsymbol{a}}$ is a $k \times k$ diagonal covariance matrix, Substituting likelihood 2.11 and prior 2.13 in equation 2.12 we get,

$$\begin{aligned} P(\boldsymbol{a}_i|\mathbf{Q}_{ii}, \boldsymbol{x}_{t-1}) = \prod_{t=2}^{T} \exp\left\{\frac{-(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})^2}{2\mathbf{Q}_{ii}}\right\} \frac{1}{(2\pi)^{1/2}(\mathbf{Q}_{ii})^{1/2}} \times \\ \exp\left\{\frac{-(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{2\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} \frac{1}{(2\pi)^{1/2}(\boldsymbol{\Sigma}_{\boldsymbol{a}ii})^{1/2}}, \end{aligned} \tag{2.14}$$

Taking out the constant term as Ka where

$$Ka = (2\pi)^{-(T+1)/2}(\mathbf{Q}_{ii})^{-1/2} \times (\boldsymbol{\Sigma}_{\boldsymbol{a}ii})^{-1/2}$$

we can re-write the above equation 2.14 as,

$$P(\boldsymbol{a}_i|\mathbf{Q}_{ii}, \boldsymbol{x}_{t-1}) = \prod_{t=2}^{T} \exp\left\{\frac{-(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1})^2}{2\mathbf{Q}_{ii}}\right\} \times \exp\left\{\frac{-(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{2\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} \times Ka, \tag{2.15}$$

Now taking the logarithm of equation 2.15 gives,

$$\log P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \mathbf{Q}) = -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1})^2}{\mathbf{Q}_{ii}}\right\} + -\frac{1}{2}\left\{\frac{(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} + \log(Ka), \tag{2.16}$$

expanding equation 2.16 we get

$$= -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{x_{i,t}^2}{\mathbf{Q}_{ii}} - \frac{2x_{i,t}\boldsymbol{a}_i\boldsymbol{x}_{t-1}}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{a}_i\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T\boldsymbol{a}_i^T}{\mathbf{Q}_{ii}}\right\} + -\frac{1}{2}\left\{\frac{\boldsymbol{a}_i^T\mathbf{I}\boldsymbol{a}_i}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} - \frac{\boldsymbol{a}_i^T\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} - \frac{\boldsymbol{\mu}_a^T\boldsymbol{a}_i}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} + \frac{\boldsymbol{\mu}_a\boldsymbol{\mu}_a^T}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} + \log(Ka), \tag{2.17}$$

we can factor out terms not involving $\boldsymbol{a}_i$ and knowing that $2x_{i,t}\boldsymbol{a}_i\boldsymbol{x}_{t-1} = 2x_{i,t}\boldsymbol{x}_{t-1}^T\boldsymbol{a}_i^T$, reduce equation 2.17 to the following,

$$= -\frac{1}{2}\sum_{t=2}^{T}\left\{-\frac{2x_{i,t}\boldsymbol{x}_{t-1}^T\boldsymbol{a}_i^T}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{a}_i\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T\boldsymbol{a}_i^T}{\mathbf{Q}_{ii}}\right\} + -\frac{1}{2}\left\{\frac{\boldsymbol{a}_i\mathbf{I}\boldsymbol{a}_i^T}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} - \frac{\boldsymbol{a}_i^T\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} - \frac{\boldsymbol{\mu}_a^T\boldsymbol{a}_i}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} + \log(Ka), \tag{2.18}$$

as we know that the product of two normal densities is also a normal distribution. A given multivariate normal distribution of any function can be written in its standard form or the canonical form [1]. We can rearrange equation 2.18 as follows:

[1]The standard form of $f(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{k}{2}}\det(\Sigma)^{\frac{1}{2}}}\exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$.
The canonical form $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \boldsymbol{C}\cdot\exp\{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Lambda}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{\eta}\}$ where, $\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$ and $\boldsymbol{\Sigma} = \boldsymbol{\Lambda}^{-1}$ with a constant $\boldsymbol{C}$.

$$\log P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \mathbf{Q}) = -\frac{1}{2}\boldsymbol{a}_i^T \left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\} \boldsymbol{a}_i + \boldsymbol{a}_i^T \left\{ \sum_{t=2}^{T} \left(\frac{x_{i,t}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\} + \log(Ka). \tag{2.19}$$

Following the canonical form we define the mean and covariance of the resulting Gaussian distribution, firstly by collecting the quadratic and the linear terms of $\boldsymbol{a}_i$ from the equation 2.19

Quadratic $\boldsymbol{\Lambda}$:

$$\left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\} \mathbf{I},$$

Linear $\boldsymbol{\eta}$:

$$\left\{ \sum_{t=2}^{T} \left(\frac{x_{i,t}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\},$$

We can define the mean, ($\tilde{\boldsymbol{\mu}}_{a_i} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$) and covariance, ($\tilde{\boldsymbol{\Sigma}}_{a_i} = \boldsymbol{\Lambda}^{-1}$) of the resulting multivariate normal distribution as:

$$\tilde{\boldsymbol{\Sigma}}_{a_i} = \left[\left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\} \mathbf{I} \right]^{-1}, \tag{2.20}$$

and

$$\tilde{\boldsymbol{\mu}}_{a_i} = \left[\left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\} \mathbf{I} \right]^{-1} \left\{ \sum_{t=2}^{T} \left(\frac{x_{i,t}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}} \right\}. \tag{2.21}$$

In this way the conditional distribution for the i^{th} row of parameter $\mathbf{A}$ is defined for the canonical form of a state space model. Treating each row of the parameter matrix $\mathbf{C}$ independently, with the i^{th} row of $\mathbf{C}$ denoted by $\boldsymbol{c}_i$ the marginal likelihood $P(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{c}_i, \mathbf{R}_{ii})$ can be given as

$$P(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{c}_i, \mathbf{R}_{ii}) = N(\boldsymbol{c}_i\boldsymbol{x}_t, \mathbf{R}_{ii}), \tag{2.22}$$

here $\mathbf{R}_{ii}$ represents the i^{th} element of a diagonal covariance matrix $\mathbf{R}$ (dimension of $\mathbf{R}$ is $k \times k$) and it also represents the variance.

By Baye's theorem, the conditional distribution $P(\mathbf{C}|\boldsymbol{y}_t, \mathbf{R}, \boldsymbol{x}_t)$ can be given as

$$P(\boldsymbol{c}_i|\boldsymbol{y}_t, \mathbf{R}_{ii}, \boldsymbol{x}_t) = \prod_{t=2}^{T} P(y_{i,t} \mid x_{i,t}, \boldsymbol{c}_i, \mathbf{R}_{ii}) P(\boldsymbol{c}_i). \tag{2.23}$$

The prior $P(\boldsymbol{c}_i)$ is defined as

$$P(\boldsymbol{c}_i) = N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_{\boldsymbol{c}}), \tag{2.24}$$

here $\boldsymbol{\mu}_c$ is $1 \times k$ mean vector and $\boldsymbol{\Sigma}_{\boldsymbol{c}}$ is a $k \times k$ diagonal covariance matrix, Substituting likelihood 2.22 and prior 2.24 in equation 2.23 we get,

$$\begin{aligned} P(\boldsymbol{c}_i|\boldsymbol{y}_t, \mathbf{R}_{ii}, \boldsymbol{x}_t) = \prod_{t=2}^{T} \exp\left\{\frac{-(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2}{2\mathbf{R}_{ii}}\right\} \frac{1}{(2\pi)^{1/2}(\mathbf{R}_{ii})^{1/2}} \times \\ \exp\left\{\frac{-(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{2\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} \frac{1}{(2\pi)^{1/2}(\boldsymbol{\Sigma}_{\boldsymbol{c}ii})^{1/2}}, \end{aligned} \tag{2.25}$$

Taking out the constant term as Kc, where

$$Kc = (2\pi)^{-(T+1)/2}(\mathbf{R}_{ii})^{-1/2} \times (\boldsymbol{\Sigma}_{\boldsymbol{c}ii})^{-1/2}$$

we can re-write equation 2.25 as,

$$\begin{aligned} P(\boldsymbol{c}_i|\boldsymbol{y}_t, \mathbf{R}_{ii}, \boldsymbol{x}_t) = \prod_{t=2}^{T} \exp\left\{\frac{-(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2}{2\mathbf{R}_{ii}}\right\} \times \\ \exp\left\{\frac{-(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{2\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} \times Kc, \end{aligned} \tag{2.26}$$

Now taking the logarithm of equation 2.26 gives,

$$\begin{aligned} \log P(\boldsymbol{c}_i|\boldsymbol{y}_t, \mathbf{R}_{ii}, \boldsymbol{x}_t) = -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2}{\mathbf{R}_{ii}}\right\} + \\ -\frac{1}{2}\left\{\frac{(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} + \log(Kc), \end{aligned} \tag{2.27}$$

expanding equation 2.27 we get

$$= -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{y_{i,t}^2}{\mathbf{R}_{ii}} - \frac{2y_{i,t}\boldsymbol{c}_i\boldsymbol{x}_t}{\mathbf{R}_{ii}} + \frac{\boldsymbol{c}_i\boldsymbol{x}_t\boldsymbol{x}_t^T\boldsymbol{c}_i^T}{\mathbf{R}_{ii}}\right\} +$$
$$-\frac{1}{2}\left\{\frac{\boldsymbol{c}_i^T\mathbf{I}\boldsymbol{c}_i}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} - \frac{\boldsymbol{c}_i^T\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} - \frac{\boldsymbol{\mu}_a^T\boldsymbol{c}_i}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} + \frac{\boldsymbol{\mu}_c\boldsymbol{\mu}_c^T}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} + \log(Kc), \tag{2.28}$$

we can factor out terms not involving $\boldsymbol{c}_i$ and knowing that $2y_{i,t}\boldsymbol{c}_i\boldsymbol{x}_t = 2y_{i,t}\boldsymbol{x}_t^T\boldsymbol{c}_i^T$, reduce equation 2.28 to the following,

$$= -\frac{1}{2}\sum_{t=2}^{T}\left\{-\frac{2y_{i,t}\boldsymbol{x}_t^T\boldsymbol{c}_i^T}{\mathbf{R}_{ii}} + \frac{\boldsymbol{c}_i\boldsymbol{x}_t\boldsymbol{x}_t^T\boldsymbol{c}_i^T}{\mathbf{R}_{ii}}\right\} +$$
$$-\frac{1}{2}\left\{\frac{\boldsymbol{c}_i\mathbf{I}\boldsymbol{c}_i^T}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} - \frac{\boldsymbol{c}_i^T\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} - \frac{\boldsymbol{\mu}_a^T\boldsymbol{c}_i}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} + \log(Kc), \tag{2.29}$$

as we know that the product of two normal densities is also a normal distribution. A given multivariate normal distribution of any function can be written in its standard form or the canonical form. We can rearrange equation 2.29 as follows:

$$\log P(\boldsymbol{c}_i|\boldsymbol{y}_t, \mathbf{R}_{ii}, \boldsymbol{x}_t) = -\frac{1}{2}\boldsymbol{c}_i^T\left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_t\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\}\boldsymbol{c}_i +$$
$$\boldsymbol{c}_i^T\left\{\sum_{t=2}^{T}\left(\frac{x_{i,t}\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{\boldsymbol{\mu}_c}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\} + \log(Kc). \tag{2.30}$$

Following the canonical form we define the mean and covariance of the resulting Gaussian distribution, firstly by collecting the quadratic and the linear terms of $\boldsymbol{c}_i$ from the equation 2.30

Quadratic $\boldsymbol{\Lambda}$:

$$\left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_t\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\}\mathbf{I},$$

Linear $\boldsymbol{\eta}$:

$$\left\{\sum_{t=2}^{T}\left(\frac{y_{i,t}\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{\boldsymbol{\mu}_c}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\},$$

We can define the mean, ($\tilde{\boldsymbol{\mu}}_{c_i} = \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}$) and covariance, ($\tilde{\boldsymbol{\Sigma}}_{c_i} = \boldsymbol{\Lambda}^{-1}$) of the resulting multivariate normal distribution as:

$$\tilde{\boldsymbol{\Sigma}}_{c_i} = \left[\left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_t\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\}\mathbf{I}\right]^{-1}, \tag{2.31}$$

and

$$\tilde{\boldsymbol{\mu}}_{c_i} = \left[\left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_t\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{1}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\}\mathbf{I}\right]^{-1}\left\{\sum_{t=2}^{T}\left(\frac{y_{i,t}\boldsymbol{x}_t^T}{\mathbf{R}_{ii}}\right) + \frac{\boldsymbol{\mu}_c}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}}\right\}. \tag{2.32}$$

Inference is performed row-wise for all the parameter matrices except the diagonal matrices $\mathbf{Q}$ and $\mathbf{R}$ for which we consider each diagonal element. Begin with the inference of error covariance parameter $\mathbf{Q}$ which is a diagonal matrix of dimension $k \times k$. The conditional distribution for each element of $\mathbf{Q}$ can be defined as

$$P(\mathbf{Q}|\{\mathbf{x}\}, \mathbf{A}) = \prod_{t=2}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{A}, \mathbf{Q})P(\mathbf{Q}) \tag{2.33}$$

Since parameter $\mathbf{Q}$ is a diagonal matrix it is more convenient to infer each diagonal element one at a time. Considering a diagonal element of $\mathbf{Q}$ as q_{ii} where $i = 1, \ldots, k$ then for each element i the prior distribution of q_{ii} can be given as an inverse gamma distribution:

$$P(q_{ii}|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} q_{ii}^{-(\alpha+1)} \exp\{-\frac{\beta}{q_{ii}}\} \tag{2.34}$$

The likelihood $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \mathbf{A}, \mathbf{Q})$ can be obtained from 2.9,

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \mathbf{A}, \mathbf{Q}) = \prod_{t=2}^{T} \frac{1}{q_{ii}^{1/2}(2\pi)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\left(\frac{x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1}}{q_{ii}^{1/2}}\right)^2\right\} \tag{2.35}$$

Now substituting equations 2.34 and 2.35 into 2.33 for each element q_{ii} we get

$$
\begin{aligned}
P(q_{ii} \mid \{\mathbf{x}\}, \boldsymbol{A}) &= \frac{\beta^{\alpha}}{\Gamma(\alpha)} q_{ii}^{-(\alpha+1)} \exp\left\{-\frac{\beta}{q_{ii}}\right\} \frac{1}{q_{ii}^{(T-1)/2}(2\pi)^{\frac{(T-1)k}{2}}} \times \\
&\quad \exp\left\{-\frac{1}{2}\sum_{t=2}^{T}\left(\frac{x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1}}{q_{ii}^{1/2}}\right)^2\right\}, \\
&= \frac{\beta^{\alpha}}{\Gamma(\alpha)} q_{ii}^{-(\alpha+1)} \exp\left\{-\frac{\beta}{q_{ii}}\right\} \frac{q_{ii}^{-(T-1)/2}}{(2\pi)^{\frac{(T-1)k}{2}}} \times \\
&\quad \exp\left\{\frac{-1/2\sum_{t=2}^{T}(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})^2}{q_{ii}}\right\}, \qquad (2.36)
\end{aligned}
$$

Taking the logarithm of equation 2.36

$$
\begin{aligned}
\log P(q_{ii} \mid \{\mathbf{x}\}, \boldsymbol{A}) &= \alpha \log(\beta) - \log(\Gamma(\alpha)) - (\alpha+1)\log(q_{ii}) - \frac{\beta}{q_{ii}} \\
&\quad -\frac{(T-1)k}{2}\log(2\pi) - \frac{(T-1)}{2}\log(q_{ii}) \\
&\quad -\frac{\frac{1}{2}\sum_{t=2}^{T}(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})^2}{q_{ii}}, \qquad (2.37)
\end{aligned}
$$

Re-arranging terms from the above equation 2.37 in the form of 2.34,

$$
\begin{aligned}
\log P(q_{ii} \mid \{\mathbf{x}\}, \boldsymbol{A}) &= \alpha \log(\beta) - \log(\Gamma(\alpha)) - \frac{(T-1)k}{2}\log(2\pi) + \\
&\quad \left\{-(\alpha+1) - \frac{(T-1)}{2}\right\}\log(q_{ii}) - \\
&\quad \left\{\frac{\beta + 1/2\sum_{t=2}^{T}(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})^2}{q_{ii}}\right\}. \qquad (2.38)
\end{aligned}
$$

The conjugate distributions of both factors in equation 2.33 of $\mathbf{Q}$ i.e. $P(\mathbf{Q})$ and $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \mathbf{A})$ follow the properties of the exponential family. Hence the product of the inverse Gamma with the exponential density will result in another exponential form of the inverse gamma distribution. In fact after rearranging equation 2.37, equation 2.38 results in an inverse gamma distribution. Therefore the coefficients of $\log(q_{ii})$ would yield $\tilde{\alpha}+1$ and the coefficient of $1/q_{ii}$ would yield $\tilde{\beta}$. Where $\tilde{\alpha}$ is

a new shape parameter and $\tilde{\beta}$ is a new scalar parameter, can be further defined as follows,

$$\tilde{\alpha} + 1 = \alpha + 1 + \frac{(T-1)}{2},$$

$$\tilde{\alpha} = \alpha + \frac{(T-1)}{2}, \tag{2.39}$$

$$\tilde{\beta} = \beta + \sum_{t=2}^{T} \frac{1}{2}(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})^2. \tag{2.40}$$

Here equations 2.39 to 2.40 shows how the conditional distribution for the error covariance matrix $\mathbf{Q}$ is calculated for a simple state space model. The error covariance matrix $\mathbf{R}$ will follow the same procedure as given for $\mathbf{Q}$.

For each element of $\mathbf{R} = r_{ii}$ the conditional distribution can be defined as,

$$P(\mathbf{R}|\mathbf{y}, \mathbf{C}) = \prod_{t=2}^{T} P(\boldsymbol{y}_t|\boldsymbol{x}_t, \mathbf{C}, \mathbf{R}) P(\mathbf{R}) \tag{2.41}$$

The prior distribution for $r_{ii} = r$ can be given as an inverse gamma distribution:

$$P(r_{ii}|\gamma, \delta) = \frac{\delta^{\gamma}}{\Gamma(\gamma)} r_{ii}^{-(\gamma+1)} \exp\{-\frac{\delta}{r_{ii}}\} \tag{2.42}$$

The likelihood $P(\boldsymbol{y}_t|\boldsymbol{x}_t, \mathbf{C}, \mathbf{R})$ can be obtained from 2.10,

$$P(\boldsymbol{y}_t|\boldsymbol{x}_t, \mathbf{C}, \mathbf{R}) = \prod_{t=2}^{T} \frac{1}{r_{ii}^{1/2}(2\pi)^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \left(\frac{y_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t}{r_{ii}^{1/2}} \right)^2 \right\} \tag{2.43}$$

Now substituting equations 2.43 and 2.42 into 2.41 for each element of r_{ii} we get

$$\begin{aligned} P(r_{ii} \mid \{\mathbf{x}\}, \boldsymbol{C}) \quad = \quad & \frac{\delta^{\gamma}}{\Gamma(\gamma)} r_{ii}^{-(\gamma+1)} \exp\{-\frac{\delta}{r_{ii}} \frac{1}{q_{ii}^{(T-1)/2}(2\pi)^{\frac{(T-1)}{2}}} \times \\ & \exp\left\{ -\frac{1}{2} \sum_{t=2}^{T} \left(\frac{y_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t}{r_{ii}^{1/2}} \right)^2 \right\}. \end{aligned} \tag{2.44}$$

Now taking the logarithm of equation 2.44,

$$\begin{aligned}\log P(r_{ii} \mid \{\mathbf{y}\}, \boldsymbol{C}) &= \gamma \log(\delta) - \log(\Gamma(\gamma)) - (\gamma+1)\log(r_{ii}) - \frac{\delta}{r_{ii}} \\ &\quad -\frac{(T-1)p}{2}\log(2\pi) - \frac{(T-1)}{2}\log(r_{ii}) \\ &\quad -\frac{\frac{1}{2}\sum_{t=2}^{T}(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2}{r_{ii}}. \end{aligned} \tag{2.45}$$

Re-arranging terms from the above equation 2.45 in the form of 2.42,

$$\begin{aligned}\log P(r_{ii} \mid \{\mathbf{y}\}, \boldsymbol{C}) &= \gamma \log(\delta) - \log(\Gamma(\gamma)) - \frac{(T-1)p}{2}\log(2\pi) \\ &\quad \left\{-(\gamma+1) - \frac{(T-1)}{2}\right\}\log(r_{ii}) - \\ &\quad \frac{\delta + 1/2\sum_{t=2}^{T}(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2)^2}{r_{ii}}. \end{aligned}$$

The product of the exponential density with the inverse gamma distribution as in 2.46 result in the inverse gamma distribution with following shape and scalar parameters:

$$\tilde{\gamma} = \gamma + 1 + \frac{(T-1)}{2} = \gamma + \frac{(T+1)}{2}, \tag{2.46}$$

$$\tilde{\delta} = \delta + \sum_{t=2}^{T}\frac{1}{2}(y_{i,t} - \boldsymbol{c}_i\boldsymbol{x}_t)^2)^2. \tag{2.47}$$

Thus each element of $\mathbf{R}_{ii}$ is inverse gamma distributed with new shape and scalar parameter $\tilde{\gamma}$ and $\tilde{\delta}$ as given above.

2.2.2 Forward Backward Gibbs Sampler

In this section we derive the conditional distribution of the hidden states $\{\mathbf{x}\}$. The conditional distribution can be defined as $P(\{\mathbf{x}\}|\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \{\mathbf{y}\})$. Sampling the posterior distribution for the state could be done in two ways, *Direct Gibbs* (the usual Gibbs sampler) [Scott, 2002] and the *forward-backward Gibbs sampler* [Scott,

2002, Chib, 1996]. In the first method, each state of the model updates on the basis of the most recent draws from its neighbours in time. However, the recursive *forward−backward Gibbs sampler* seems to be more convenient for the SSM.

Let us assume that $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}\}$ is the set of all parameters and $P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{y}_{1:T}, \boldsymbol{\theta})$ can be factorised as follows.

$$P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \propto P(\boldsymbol{x}_{t-1} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \mathbf{y}_{t-1}, \boldsymbol{\theta}) \tag{2.48}$$

where $P(\boldsymbol{x}_{t-1}|\mathbf{y}_{1:T}, \boldsymbol{\theta})$ can be calculated by using the forward algorithm (i.e. Kalman filtering) with some initial mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{v}_t$.

$$P(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:T}, \boldsymbol{\theta}) \sim N(\boldsymbol{x}_{t-1}|\boldsymbol{\mu}_t, \boldsymbol{v}_t). \tag{2.49}$$

We recursively calculate the mean and covariance matrix of $P(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:T})$ and estimates at time $t-1$. $P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{\theta})$ also follows a Gaussian distribution with mean and covariance given by equation 2.7 i.e.

$$P(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \propto N(\mathbf{A}\boldsymbol{x}_{t-1}, \mathbf{Q}) \tag{2.50}$$

Substituting equations 2.49 and 2.50 into 2.48 we can obtain the posterior distribution for the states as follows

$$\begin{aligned}
P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}) &\sim N(\boldsymbol{x}_{t-1} \mid \boldsymbol{\mu}_t, \boldsymbol{v}_t) N(\boldsymbol{x}_t \mid \mathbf{A}\boldsymbol{x}_{t-1}, \mathbf{Q}), \\
&\propto \exp\{-\frac{1}{2}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_t)^T \boldsymbol{v}_t^{-1}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_t)\} \times \\
&\quad \exp\{-\frac{1}{2}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1})^T \mathbf{Q}^{-1}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1})\}, \\
&\propto \exp\{-\frac{1}{2}\{(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_t)^T \boldsymbol{v}_t^{-1}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}_t) + (\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1}) \times \\
&\quad \mathbf{Q}^{-1} \times (\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1})\},
\end{aligned} \tag{2.51}$$

After taking out the factor of -2 and constants resulting from equation 2.51 as Kx

we take the logarithm of equation 2.51:

$$\begin{aligned} -2\log(P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t;\boldsymbol{y}_{1:T},\boldsymbol{\theta})) &= \{(\boldsymbol{x}_{t-1}-\boldsymbol{\mu}_t)^T\boldsymbol{v}_t^{-1}(\boldsymbol{x}_{t-1}-\boldsymbol{\mu}_t)+(\boldsymbol{x}_t-\mathbf{A}\boldsymbol{x}_{t-1})^T\times \\ &\quad Q^{-1}(\boldsymbol{x}_t-\mathbf{A}\boldsymbol{x}_{t-1})\}+\log Kx, \qquad (2.52) \\ &= \{\boldsymbol{x}_{t-1}^T\boldsymbol{v}_t^{-1}\boldsymbol{x}_{t-1}-\boldsymbol{x}_{t-1}^T\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t-\boldsymbol{\mu}_t^T\boldsymbol{v}_t^{-1}\boldsymbol{x}_{t-1}+ \\ &\quad \boldsymbol{\mu}_t^T\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t+\boldsymbol{x}_tQ^{-1}\boldsymbol{x}_t-\boldsymbol{x}_tQ^{-1}\mathbf{A}\boldsymbol{x}_{t-1}- \\ &\quad \boldsymbol{x}_{t-1}^T\mathbf{A}^TQ^{-1}\boldsymbol{x}_t+\boldsymbol{x}_{t-1}^T\mathbf{A}^TQ^{-1}\mathbf{A}\boldsymbol{x}_{t-1}\}+\log Kx, \end{aligned}$$

Factor out terms not including $\boldsymbol{x}_{t-1}$,

$$\begin{aligned} -2\log(P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t;\boldsymbol{y}_{1:T},\boldsymbol{\theta})) &= \{\boldsymbol{x}_{t-1}^T\boldsymbol{v}_t^{-1}\boldsymbol{x}_{t-1}-2\boldsymbol{x}_{t-1}^T\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t-2\boldsymbol{x}_{t-1}^T\mathbf{A}^TQ^{-1}\boldsymbol{x}_t+ \\ &\quad \boldsymbol{x}_{t-1}^T\mathbf{A}^TQ^{-1}\mathbf{A}\boldsymbol{x}_{t-1}\}+\log Kx. \qquad (2.53) \end{aligned}$$

The above equation can be rearraged by taking factor of 2 as

$$\begin{aligned} -2\log(P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t;\boldsymbol{y}_{1:T},\boldsymbol{\theta})) &= -\frac{1}{2}\boldsymbol{x}_{t-1}^T(\boldsymbol{v}_t^{-1}+\mathbf{A}^TQ^{-1}\mathbf{A})\boldsymbol{x}_{t-1}+ \\ &\quad \boldsymbol{x}_{t-1}^T(\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t+\mathbf{A}^TQ^{-1}\boldsymbol{x}_t)+\log Kx. \qquad (2.54) \end{aligned}$$

Expanding equation 2.54 as shown above and by collecting the linear and quadratic terms of the state $\boldsymbol{x}_{t-1}$, we obtain a new mean, $\boldsymbol{\mu}_x$, and covariance, $\boldsymbol{\sigma}_x$, for the state $\boldsymbol{x}_{t-1}$:

Quadratic term: $\boldsymbol{x}_{t-1}^T\underbrace{\{\boldsymbol{v}_t^{-1}+\frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}}_{\Lambda}\boldsymbol{x}_{t-1}$,

Linear term: $\boldsymbol{x}_{t-1}\underbrace{\{\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t^T+\frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\boldsymbol{x}_t\}}_{\eta}$.

Therefore the mean, μ_x and covariance, σ_x, can be written as:

$$\begin{aligned} \boldsymbol{\Sigma}_x &= \boldsymbol{\Lambda}^{-1}, \\ &= \{\boldsymbol{v}_t^{-1}+\frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}^{-1}. \qquad (2.55) \end{aligned}$$

$$\begin{aligned}\boldsymbol{\mu}_x &= \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}, \\ &= \{\boldsymbol{v}_t^{-1} + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}^{-1}\{\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t^T + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\boldsymbol{x}_t\}. \end{aligned} \tag{2.56}$$

Therefore $P(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t; \boldsymbol{y}_{1:T}, \boldsymbol{\theta})) \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$. The pseudocode for the Gibbs sampler algorithm 1 iterates between two steps; firstly sample all parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}\}$. Secondly, sample the states $\{\mathbf{x}\}$ by forward filtering and backward sampling. An optional last step may also be used to estimate observations for missing time points. We will use this step in the forthcoming experiments for the validation of the MCMC algorithm.

Algorithm 1: Gibbs Sampler algorithm for the cannonical SSM.

Input: Randomly initialize parameters $\boldsymbol{\mu}_A$, $\boldsymbol{\Sigma}_A$, $\mathbf{Q}$, μ_C, $\boldsymbol{\Sigma}_C$, $\mathbf{R}$, α, β, γ, δ, and the latent variable $\{\mathbf{x}\}$. Fix the length of MCMC chain N.

1 Here index k indicates number of rows in the state sequence and p is the number of rows in observation sequence.

Output: N samples of $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{x}$

2 For each row $i = 1, \ldots, k$

3 Sample $\boldsymbol{a}_i|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{Q} \sim N(\tilde{\boldsymbol{\mu}}_{a_i}, \tilde{\boldsymbol{\Sigma}}_{a_i})$ using 2.20 and 2.21

4 Update $\mathbf{A}$

5 Sample $\boldsymbol{q}_{ii}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{A} \sim IG(\tilde{\alpha}, \tilde{\beta})$ using 2.39 and 2.40

6 Update $\mathbf{Q}$

7 For each row $s = 1, \ldots, p$

8 Sample $\boldsymbol{c}_s|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{R} \sim N(\tilde{\boldsymbol{\mu}}_{c_i}, \tilde{\boldsymbol{\Sigma}}_{c_i})$ using 2.31 and 2.32

9 Update $\mathbf{C}$

10 Sample $\boldsymbol{r}_{ss}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{C} \sim IG(\tilde{\gamma}, \tilde{\delta})$ using 2.46 and 2.47

11 Update $\mathbf{R}$

12 Forward sampling: for $t = 1, \ldots, T$

13 Sample $\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R} \sim N(\boldsymbol{\mu}_t, \boldsymbol{v}_t)$ using Kalman filtering

14 Backward sampling: for $t = (T-1), \ldots, 1$

15 Sample $\mathbf{x}_{t-1}|\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \mathbf{y}_{1:T}, \mathbf{x}_t \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ using 2.55

16 Update $\{\mathbf{x}\}$

17 Repeat steps 2-14 untill N samples of $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, $\mathbf{R}$, $\{\mathbf{x}\}$ collected.

The collected samples from the Gibbs sampler algorithm, 1, for the parameters $\mathbf{A}$, $\mathbf{C}$, $\mathbf{Q}$, $\mathbf{R}$ and $\{\mathbf{x}\}$ will be further investigated with convergence diagnostics as described in Section 2.3.

2.2.3 State Space model with Feedback

We now consider the state space model with feedback given by equations 2.1 and 2.2. The joint probability distribution function for T states and observations can be defined as,

$$P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x}_1)P(\mathbf{y}_1|\mathbf{x}_1, \mathbf{y}_0)\prod_{t=2}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1})P(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{t-1}) \tag{2.57}$$

where indices $t = 1, \cdots, T$ represent the time steps.

The conditional distribution of the states and the observables is assumed to be Gaussian and given by

$$\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1} \sim N(\mathbf{A}\boldsymbol{x}_{t-1} + \mathbf{B}\boldsymbol{y}_{t-1}, \mathbf{Q}), \tag{2.58}$$

$$\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1} \sim N(\mathbf{C}\boldsymbol{x}_t + \mathbf{D}\boldsymbol{y}_{t-1}, \mathbf{R}). \tag{2.59}$$

Hence, the multivariate density functions for a given set of observations in equations 2.58 and 2.59 can be written as,

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1} - \mathbf{B}\boldsymbol{y}_{t-1})^T\mathbf{Q}^{-1}(\boldsymbol{x}_t - \mathbf{A}\boldsymbol{x}_{t-1} - \mathbf{B}\boldsymbol{y}_{t-1})\}}{(2\pi)^{\frac{k}{2}}\det(\mathbf{Q})^{\frac{1}{2}}}, \tag{2.60}$$

$$P(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}) = \frac{\exp\{-\frac{1}{2}(\boldsymbol{y}_t - \mathbf{C}\boldsymbol{x}_t - \mathbf{D}\boldsymbol{y}_{t-1})^T\mathbf{R}^{-1}(\boldsymbol{y}_t - \mathbf{C}\boldsymbol{x}_t - \mathbf{D}\boldsymbol{y}_{t-1})\}}{(2\pi)^{\frac{p}{2}}\det(\mathbf{R})^{\frac{1}{2}}}. \tag{2.61}$$

where det denotes the determinant and $P(\mathbf{x}_1)$ is assumed to be Gaussian as defined by equation 2.6.

The Gibbs Sampler for an SSM with feed back (as given in equations 2.1 and 2.2), proceeds by drawing from the complete full-conditional distributions of the parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$ given the states $\{\mathbf{x}\}$ and $\{\mathbf{y}\}$. Following

the sampling of the parameters we draw from the complete full conditional distribution of hidden state $\{\mathbf{x}\}$ for time steps $t = 1, \cdots, T$ given the parameters and the observations $\{\mathbf{y}\}$, according to the following:

$$
\begin{aligned}
P(\mathbf{A}|\mathbf{B}, \mathbf{Q}, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \mathbf{B}, \mathbf{Q})P(\mathbf{A}) && (2.62)\\
P(\mathbf{B}|\mathbf{A}, \mathbf{Q}, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \mathbf{A}, \boldsymbol{Q})P(\mathbf{B}) && (2.63)\\
P(\mathbf{Q}|\mathbf{A}, \mathbf{B}, \mathbf{x}_{t-1}, \mathbf{y}_{t-1}) &= P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \mathbf{B}, \mathbf{A})P(\mathbf{Q}) && (2.64)\\
P(\mathbf{C}|\boldsymbol{D}, \mathbf{R}, \mathbf{x}_t, \mathbf{y}_{t-1}) &= P(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \mathbf{D}, \boldsymbol{R})P(\mathbf{C}) && (2.65)\\
P(\mathbf{D}|\mathbf{A}, \mathbf{R}, \mathbf{x}_t, \mathbf{y}_{t-1}) &= P(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \mathbf{C}, \boldsymbol{R})P(\mathbf{D}) && (2.66)\\
P(\mathbf{R}|\mathbf{C}, \mathbf{D}, \mathbf{x}_t, \mathbf{y}_{t-1}) &= P(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \mathbf{C}, \mathbf{D})P(\mathbf{R}) && (2.67)\\
P(\mathbf{x}|\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{y}_{1:T}) &= P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{A}, \mathbf{B})P(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}, \mathbf{A}, \mathbf{B}) && (2.68)
\end{aligned}
$$

Firstly we derive the conditional distribution for the dynamic set of parameters of the hidden states i.e. $\mathbf{A}$ and $\mathbf{B}$. Assuming the parameter $\mathbf{B}$ is known (or initialised randomly) we can derive the conditional distribution of $\mathbf{A}$ as follows. Considering each row of the parameter matrix $\mathbf{A}$ and $\mathbf{B}$ independently, for the i^{th} row of $\mathbf{A}$ and $\mathbf{B}$ denoted by $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ respectively,

$$
P(\boldsymbol{a}_i|\boldsymbol{b}_i, \mathbf{Q}, \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}) = \prod_{t=2}^{T} P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{a}_i, \boldsymbol{b}_i, \mathbf{Q}_{ii})P(\boldsymbol{a}_i). \tag{2.69}
$$

Following equation 2.58 and considering a Gussian prior $\boldsymbol{a}_i \sim N(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$ we can define LHS of equation 2.69 as,

$$
\begin{aligned}
P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{b}_i, \mathbf{Q}_{ii}) &= \prod_{t=2}^{T} N(\boldsymbol{x}_t - \boldsymbol{a}_i\boldsymbol{x}_{t-1} - \boldsymbol{b}_i\boldsymbol{y}_{t-1}, \mathbf{Q}_{ii})N(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)\\
&= \prod_{t=2}^{T} \exp\left\{\frac{-(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1} - \boldsymbol{b}_i\boldsymbol{y}_{t-1})^2}{2\mathbf{Q}_{ii}}\right\} \frac{1}{(2\pi)^{1/2}(\mathbf{Q}_{ii})^{1/2}} \times\\
&\qquad \exp\left\{\frac{-(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{2\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} \frac{1}{(2\pi)^{1/2}(\boldsymbol{\Sigma}_{\boldsymbol{a}ii})^{1/2}}, \qquad (2.70)
\end{aligned}
$$

Taking out the constant term as Ka, we can re-write the above equation as,

$$P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{b}_i, \mathbf{Q}_{ii}) = \prod_{t=2}^{T} \exp\left\{\frac{-(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{Q}_{ii}}\right\} \times \exp\left\{\frac{-(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{2\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} \times Ka, \tag{2.71}$$

Now taking the logarithm of above equation gives,

$$\log P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{b}_i, \mathbf{Q}_{ii}) = -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2}{\mathbf{Q}_{ii}}\right\} + -\frac{1}{2}\left\{\frac{(\boldsymbol{a}_i - \boldsymbol{\mu}_a)^T(\boldsymbol{a}_i - \boldsymbol{\mu}_a)}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} + \log(Ka), \tag{2.72}$$

expanding equation 2.72 and factor out terms not involving $\boldsymbol{a}_i$ can reduce equation 2.72 to the following, (this can done in exact manner following steps shown in the canonical SSMs from section 2.1)

$$\log P(\boldsymbol{a}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{b}_i, \mathbf{Q}_{ii}) = -\frac{1}{2}\boldsymbol{a}_i^T\left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}}\right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\}\boldsymbol{a}_i + \boldsymbol{a}_i^T\left\{\sum_{t=2}^{T}\frac{(x_{i,t} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\} + \log(Ka), \tag{2.73}$$

Collecting quadratic and linear term of $\boldsymbol{a}_i$ from equation 2.73 we get,

$$\tilde{\boldsymbol{\Sigma}}_{a_i} = \left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}}\right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\}^{-1}, \tag{2.74}$$

$$\tilde{\boldsymbol{\mu}}_{a_i} = \left\{\sum_{t=2}^{T}\left(\frac{\boldsymbol{x}_{t-1}\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}}\right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\}^{-1}\left\{\sum_{t=2}^{T}\frac{(x_{i,t} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})\boldsymbol{x}_{t-1}^T}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{\mu}_a}{\boldsymbol{\Sigma}_{\boldsymbol{a}ii}}\right\}. \tag{2.75}$$

Similarly following the conditional distribution 2.63 for the i^{th} row of $\mathbf{B}$ can be given as

$$P(\boldsymbol{b}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = \prod_{t=2}^{T} P(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1}, \boldsymbol{a}_i, \boldsymbol{b}_i, \mathbf{Q}_{ii}) P(\boldsymbol{b}_i). \tag{2.76}$$

Substituting equation 2.58 into 2.76 and using a Gussian prior $\boldsymbol{b}_i \sim N(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$ we get

$$P(\boldsymbol{b}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = \prod_{t=2}^{T} N(\boldsymbol{x}_t - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1}, \mathbf{Q}) N(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a),$$
$$= \prod_{t=2}^{T} \exp\left\{ \frac{-(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{Q}_{ii}} \right\} \frac{1}{(2\pi)^{1/2} (\mathbf{Q}_{ii})^{1/2}} \times$$
$$\exp\left\{ \frac{-(\boldsymbol{b}_i - \boldsymbol{\mu}_b)^T (\boldsymbol{b}_i - \boldsymbol{\mu}_b)}{2\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\} \frac{1}{(2\pi)^{1/2} (\boldsymbol{\Sigma}_{\boldsymbol{b}ii})^{1/2}}, \qquad (2.77)$$

Taking out the constant term as Kb, we can re-write the above equation as,

$$P(\boldsymbol{b}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = \prod_{t=2}^{T} \exp\left\{ \frac{-(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{Q}_{ii}} \right\} \times$$
$$\prod_{i=1}^{k} \exp\left\{ \frac{-(\boldsymbol{b}_i - \boldsymbol{\mu}_b)^T (\boldsymbol{b}_i - \boldsymbol{\mu}_b)}{2\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\} \times Kb, \qquad (2.78)$$

Now taking the logarithm of above equation gives,

$$\log P(\boldsymbol{b}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = -\frac{1}{2} \sum_{t=2}^{T} \left\{ \frac{(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2}{\mathbf{Q}_{ii}} \right\} +$$
$$-\frac{1}{2} \left\{ \frac{(\boldsymbol{b}_i - \boldsymbol{\mu}_b)^T (\boldsymbol{b}_i - \boldsymbol{\mu}_b)}{2\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\} + \log(Kb), \qquad (2.79)$$

expanding equation 2.79 and factor out terms not involving $\boldsymbol{b}_i$ can reduce equation 2.79 to the following,

$$\log P(\boldsymbol{b}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{a}_i, \mathbf{Q}_{ii}) = -\frac{1}{2} \boldsymbol{b}_i^T \left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{y}_{t-1} \boldsymbol{y}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\} \boldsymbol{b}_i +$$
$$\boldsymbol{b}_i^T \left\{ \sum_{t=2}^{T} \frac{(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1}) \boldsymbol{y}_{t-1}^T}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{\mu}_b}{\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\} + \log(Kb), \qquad (2.80)$$

Collecting quadratic and linear term of $\boldsymbol{a}_i$ from equation 2.80 we get,

$$\tilde{\boldsymbol{\Sigma}}_{b_i} = \left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{y}_{t-1} \boldsymbol{y}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\}^{-1}, \qquad (2.81)$$

$$\tilde{\boldsymbol{\mu}}_{b_i} = \left\{ \sum_{t=2}^{T} \left(\frac{\boldsymbol{y}_{t-1}\boldsymbol{y}_{t-1}^T}{\mathbf{Q}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\}^{-1} \boldsymbol{b}_i^T \left\{ \sum_{t=2}^{T} \frac{(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1})\boldsymbol{y}_{t-1}^T}{\mathbf{Q}_{ii}} + \frac{\boldsymbol{\mu}_b}{\boldsymbol{\Sigma}_{\boldsymbol{b}ii}} \right\}. \quad (2.82)$$

In this way, we derive the conditional distribution for the dynamic set of parameters of the hidden states i.e. **A** and **B**. Assuming the parameter **D** is known (or initialised randomly) we can derive the conditional distribution of **C** as follows, considering each i^{th} row of matrix **C** denoted as $\boldsymbol{c}_i$ and **D** as $\boldsymbol{d}_i$,

$$P(\boldsymbol{c}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{d}_i, \mathbf{R}_{ii}) = \prod_{t=2}^{T} P(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \boldsymbol{c}_i, \boldsymbol{d}_i, \mathbf{R}_{ii}) P(\boldsymbol{c}_i). \quad (2.83)$$

Substituting equation 2.58 into 2.83 and using a Gussian prior $\boldsymbol{c}_i \sim N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ we get

$$\begin{aligned} P(\boldsymbol{c}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{d}_i, \mathbf{R}_{ii}) &= \prod_{t=2}^{T} N(\boldsymbol{y}_t - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1}, \mathbf{R}_{ii}) N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \\ &= \prod_{t=2}^{T} \exp\left\{ \frac{-(\boldsymbol{y}_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{R}_{ii}} \right\} \frac{1}{(2\pi)^{1/2}(\mathbf{R}_{ii})^{1/2}} \times \\ &\qquad \exp\left\{ \frac{-(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{2\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\} \frac{1}{(2\pi)^{1/2}(\boldsymbol{\Sigma}_{\boldsymbol{c}ii})^{1/2}}, \end{aligned} \quad (2.84)$$

Taking out the constant term as Kc we can re-write the above equation as,

$$\begin{aligned} P(\boldsymbol{c}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{d}_i, \mathbf{R}_{ii}) &= \prod_{t=2}^{T} \exp\left\{ \frac{-(\boldsymbol{y}_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{R}_{ii}} \right\} \times \\ &\qquad \exp\left\{ \frac{-(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{2\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\} \times Kc, \end{aligned} \quad (2.85)$$

Now taking the logarithm of above equation gives,

$$\begin{aligned} \log P(\boldsymbol{c}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{d}_i, \mathbf{R}_{ii}) &= -\frac{1}{2} \sum_{t=1}^{T} \left\{ \frac{(\boldsymbol{y}_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{\mathbf{R}_{ii}} \right\} + \\ &\qquad -\frac{1}{2} \left\{ \frac{(\boldsymbol{c}_i - \boldsymbol{\mu}_c)^T(\boldsymbol{c}_i - \boldsymbol{\mu}_c)}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\} + \log(Kc), \end{aligned} \quad (2.86)$$

expanding equation 2.86 and factor out terms not involving $\boldsymbol{c}_i$ can reduce equation 2.86 to the following,

$$\log P(\boldsymbol{c}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{d}_i, \mathbf{R}_{ii}) = -\frac{1}{2}\boldsymbol{c}_i^T \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{x}_t \boldsymbol{x}_t^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\} \boldsymbol{c}_i + \boldsymbol{c}_i^T \left\{ \sum_{t=1}^{T} \frac{(\boldsymbol{y}_{i,t} - \boldsymbol{d}_i \boldsymbol{y}_{t-1})\boldsymbol{x}_t^T}{\mathbf{R}_{ii}} + \frac{\boldsymbol{\mu}_c}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\} + \log(Kc), \tag{2.87}$$

Collecting quadratic and linear term of $\boldsymbol{c}_i$ from equation 2.87 we get,

$$\tilde{\boldsymbol{\Sigma}}_{c_i} = \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{x}_t \boldsymbol{x}_t^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\}^{-1}, \tag{2.88}$$

$$\tilde{\boldsymbol{\mu}}_{c_i} = \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{x}_t \boldsymbol{x}_t^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\}^{-1} \left\{ \sum_{t=1}^{T} \frac{(\boldsymbol{y}_{i,t} - \boldsymbol{d}_i \boldsymbol{y}_{t-1})\boldsymbol{x}_t^T}{\mathbf{R}_{ii}} + \frac{\boldsymbol{\mu}_c}{\boldsymbol{\Sigma}_{\boldsymbol{c}ii}} \right\}. \tag{2.89}$$

Similary following the conditional distribution 2.66 for the i^{th} row of $\mathbf{D}$ can be given as

$$P(\boldsymbol{d}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{c}_i, \mathbf{R}_{ii}) = \prod_{t=2}^{T} P(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{y}_{t-1}, \boldsymbol{c}_i, \boldsymbol{d}_i, \mathbf{R}_{ii}) P(\boldsymbol{d}_i). \tag{2.90}$$

Substituting equation 2.58 into 2.90 and using a Gussian prior $\boldsymbol{d}_i \sim N(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$ we get

$$\begin{aligned} P(\boldsymbol{d}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{c}_i, \mathbf{R}_{ii}) &= \prod_{t=2}^{T} N(\boldsymbol{y}_t - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1}, \mathbf{R}_{ii}) N(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d), \\ &= \prod_{t=2}^{T} \exp\left\{ \frac{-(x_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{R}_{ii}} \right\} \frac{1}{(2\pi)^{1/2} (\mathbf{R}_{ii})^{1/2}} \times \\ &\qquad \exp\left\{ \frac{-(\boldsymbol{d}_i - \boldsymbol{\mu}_d)^T (\boldsymbol{d}_i - \boldsymbol{\mu}_d)}{2\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\} \frac{1}{(2\pi)^{1/2} (\boldsymbol{\Sigma}_{\boldsymbol{d}ii})^{1/2}}, \end{aligned} \tag{2.91}$$

Taking out the constant term as Kd we can re-write the above equation as,

$$P(\boldsymbol{d}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{c}_i, \mathbf{R}_{ii}) = \prod_{t=2}^{T} \exp\left\{ \frac{-(x_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{2\mathbf{R}_{ii}} \right\} \times \exp\left\{ \frac{-(\boldsymbol{d}_i - \boldsymbol{\mu}_d)^T (\boldsymbol{d}_i - \boldsymbol{\mu}_d)}{2\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\} \times Kd, \tag{2.92}$$

Now taking the logarithm of above equation gives,

$$\log P(\boldsymbol{d}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{c}_i, \mathbf{R}_{ii}) = -\frac{1}{2} \sum_{t=1}^{T} \left\{ \frac{(x_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2}{\mathbf{R}_{ii}} \right\} + -\frac{1}{2} \left\{ \frac{(\boldsymbol{d}_i - \boldsymbol{\mu}_d)^T (\boldsymbol{d}_i - \boldsymbol{\mu}_d)}{2\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\} + \log(Kd), \tag{2.93}$$

expanding equation 2.93 and factor out terms not involving $\boldsymbol{d}_i$ can reduce equation 2.93 to the following,

$$\log P(\boldsymbol{d}_i \mid \{\mathbf{x}\}, \{\mathbf{y}\}, \boldsymbol{c}_i, \mathbf{R}_{ii}) = -\frac{1}{2} \boldsymbol{d}_i^T \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{y}_{t-1} \boldsymbol{y}_{t-1}^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\} \boldsymbol{d}_i + \boldsymbol{d}_i^T \left\{ \sum_{t=1}^{T} \frac{(x_{i,t} - \boldsymbol{c}_i^T \boldsymbol{x}_t) \boldsymbol{y}_{t-1}^T}{\mathbf{R}_{ii}} + \frac{\boldsymbol{\mu}_d}{\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\} + \log(Kd), \tag{2.94}$$

Collecting quadratic and linear term of $\boldsymbol{d}_i$ from equation 2.94 we get,

$$\tilde{\boldsymbol{\Sigma}}_{d_i} = \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{y}_{t-1} \boldsymbol{y}_{t-1}^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\}^{-1}, \tag{2.95}$$

and

$$\tilde{\boldsymbol{\mu}}_{d_i} = \left\{ \sum_{t=1}^{T} \left(\frac{\boldsymbol{y}_{t-1} \boldsymbol{y}_{t-1}^T}{\mathbf{R}_{ii}} \right) + \frac{\mathbf{I}}{\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\}^{-1} \left\{ \sum_{t=1}^{T} \frac{(x_{i,t} - \boldsymbol{c}_i^T \boldsymbol{x}_t) \boldsymbol{y}_{t-1}^T}{\mathbf{R}_{ii}} + \frac{\boldsymbol{\mu}_d}{\boldsymbol{\Sigma}_{\boldsymbol{d}ii}} \right\}. \tag{2.96}$$

The error covariance parameter $\mathbf{Q}$ is a $k \times k$ dimensional diagonal matrix.

The conditional distribution for each element of $\mathbf{Q}$ can be defined as

$$P(\mathbf{Q}|\{\mathbf{x}\},\{\mathbf{y}\},\mathbf{A},\mathbf{B}) = \prod_{t=2}^{T} P(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_{t-1},\mathbf{A},\mathbf{B},\mathbf{Q})P(\mathbf{Q}). \tag{2.97}$$

Since parameter $\mathbf{Q}$ is a diagonal matrix it is more convenient to infer each of the diagonal elements one at a time. Considering a diagonal element of $\mathbf{Q}$ as q_{ii} where $i = 1,\ldots,k$, then for each element i the distribution of $q_{ii} = q$ can be given as:

$$P(q_{ii}|\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} q_{ii}^{-(\alpha+1)} \exp\{-\frac{\beta}{q_{ii}}\} \tag{2.98}$$

The likelihood $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\mathbf{y}_{t-1},\mathbf{A},\mathbf{B},\mathbf{Q})$ can be obtained from 2.58,

$$P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\mathbf{y}_{t-1},\mathbf{A},\mathbf{B},\mathbf{Q}) = \prod_{i=1}^{k}\prod_{t=2}^{T} \frac{\exp\{-\frac{1}{2}(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1} - \boldsymbol{b}_i\boldsymbol{y}_{t-1})^2\}}{(2\pi)^{\frac{k}{2}} q_{ii}^{\frac{1}{2}}} \tag{2.99}$$

Now substituting equation 2.34 and 2.35 into 2.33 for each i^{th} element of q_{ii} we get

$$P(q_{ii}|\{\mathbf{x}\},\{\mathbf{y}\},\boldsymbol{a}_i,\boldsymbol{b}_i) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} q_{ii}^{-(\alpha-1)} \exp\left\{-\frac{\beta}{q_{ii}}\right\}$$
$$\frac{\exp\{-\sum_{t=2}^{T}(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1} - \boldsymbol{b}_i\boldsymbol{y}_{t-1})^2/2q_{ii}\}}{(2\pi)^{\frac{(T-1)k}{2}} q_{ii}^{\frac{(T-1)}{2}}}, \tag{2.100}$$

Taking the logarithm of equation 2.100

$$\begin{aligned} P(q_{ii}|\{\mathbf{x}\},\{\mathbf{y}\},\boldsymbol{a}_i,\boldsymbol{b}_i) &= \alpha\log(\beta) - \log(\Gamma(\alpha)) - (\alpha-1)\log(q_{ii}) - \frac{\beta}{q_{ii}} \\ &\quad -\frac{1}{2}\sum_{t=2}^{T}\left\{\frac{(x_{i,t} - \boldsymbol{a}_i\boldsymbol{x}_{t-1} - \boldsymbol{b}_i\boldsymbol{y}_{t-1})^2}{q_{ii}}\right\} \\ &\quad -\frac{(T-1)k}{2}\log(2\pi) - \frac{(T-1)}{2}\log(q_{ii}). \end{aligned} \tag{2.101}$$

The conjugate distributions of both factors in equation 2.33 of $\mathbf{Q}$ i.e. $P(\mathbf{Q})$ and $P(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\mathbf{y}_{t-1},\mathbf{A},\mathbf{B},\mathbf{Q})$ follow the properties of the exponential family. Hence the product of the inverse Gamma with the exponential density will result in an-

other exponential form of the inverse gamma distribution. This follows simply by collecting the coefficients of $\log(q_{ii})$ and $1/q_{ii}$ from the above equation 2.37.

Therefore each element of $\mathbf{Q}$ (i.e. q_{ii}) is inverse gamma distributed with a new shape and scalar parameter $\tilde{\alpha}$ and $\tilde{\beta}$, respectively

$$\tilde{\alpha} = \alpha + 1 + \frac{(T-1)}{2} = \alpha + \frac{(T+1)}{2}, \tag{2.102}$$

$$\tilde{\beta} = \beta + \sum_{t=2}^{T} \frac{1}{2}(x_{i,t} - \boldsymbol{a}_i \boldsymbol{x}_{t-1} - \boldsymbol{b}_i \boldsymbol{y}_{t-1})^2. \tag{2.103}$$

Therefore equations 2.33 to 2.40 show how the conditional distribution for the error covariance matrix $\mathbf{Q}$ is calculated for a state space model with feedback. The error covariance matrix $\mathbf{R}$ will follow the same procedure as given for $\mathbf{Q}$,

$$\tilde{\gamma} = \gamma + 1 + \frac{(T-1)}{2} = \gamma + \frac{(T+1)}{2}, \tag{2.104}$$

$$\tilde{\delta} = \delta + \sum_{t=1}^{T} \frac{1}{2}(x_{i,t} - \boldsymbol{c}_i \boldsymbol{x}_t - \boldsymbol{d}_i \boldsymbol{y}_{t-1})^2. \tag{2.105}$$

2.2.4 Forward Backward Gibbs Sampler for the SSM with feedback

As in section 2.2.2, we assume that $\boldsymbol{\theta} = \{\mathbf{A},\ \mathbf{B},\ \mathbf{C},\ \mathbf{D},\ \mathbf{Q},\ \mathbf{R}\}$ is the set of all parameters and $P(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_{1:T}, \boldsymbol{\theta})$ can be factorised as follows

$$P(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_{1:T}, \boldsymbol{\theta}) \propto P(\mathbf{x}_{t-1} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}) P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \boldsymbol{\theta}), \tag{2.106}$$

where $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:T}, \boldsymbol{\theta})$ can be calculated by using the forward algorithm (i.e. Kalman filtering) with some initial mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{v}_t$.

$$P(\mathbf{x}_{t-1}|\mathbf{y}_{1:T}, \boldsymbol{\theta}) \sim N(\mathbf{x}_{t-1}|\boldsymbol{\mu}_t, \boldsymbol{v}_t) \tag{2.107}$$

We recursively calculate the mean and covariance matrix of $P(\mathbf{x}_{t-1}|\mathbf{y}_{1:T})$ and estimates at time $t-1$.

$P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\theta})$ also follows a Gaussian distribution with mean and covariance given by equation 2.58,

$$P(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \boldsymbol{\theta}) = N(\mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{y}_{t-1}, \mathbf{Q}) \tag{2.108}$$

Substituting equations 2.107 and 2.108 into 2.106 we can define the posterior distribution for the states as follows

$$\begin{aligned} P(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}_{1:T}, \boldsymbol{\theta}) &= N(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}_t, \boldsymbol{v}_t) N(\mathbf{x}_t \mid \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{y}_{t-1}, \mathbf{Q}), \\ &\propto \exp\{-\frac{1}{2}(\mathbf{x}_{t-1} - \boldsymbol{\mu}_t)' \boldsymbol{v}_t^{-1} (\mathbf{x}_{t-1} - \boldsymbol{\mu}_t)\} \times \\ &\quad \exp\{-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})' \mathbf{Q}^{-1} (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})\}, \\ &\propto \exp\{-\frac{1}{2}\{(\mathbf{x}_{t-1} - \boldsymbol{\mu}_t)' \boldsymbol{v}_t^{-1} (\mathbf{x}_{t-1} - \boldsymbol{\mu}_t) + (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1}) \times \\ &\quad \mathbf{Q}^{-1} \times (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})^T\}. \end{aligned} \tag{2.109}$$

After taking out the factor of -2 and constant as Kx resulting from equation 2.109 we take the logarithm of equation 2.109:

$$\begin{aligned} -2\log(P(\mathbf{x}_{t-1}|\mathbf{x}_t; \mathbf{y}_{1:T}, \boldsymbol{\theta})) &= \{(\mathbf{x}_{t-1} - \boldsymbol{\mu}_t)^T \boldsymbol{v}_t^{-1} (\mathbf{x}_{t-1} - \boldsymbol{\mu}_t) + \times \\ &\quad (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})^T Q^{-1} (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})\} + \\ &\quad \log(Kx). \end{aligned} \tag{2.110}$$

Expanding and rearranging equation 2.110 as

$$\begin{aligned} -2\log(P(\mathbf{x}_{t-1}|\mathbf{x}_t; \mathbf{y}_{1:T}, \boldsymbol{\theta})) &= -\frac{1}{2}\mathbf{x}_{t-1}^T \{\boldsymbol{v}_t^{-1} + \frac{1}{2}\mathbf{A}^T \mathbf{Q}^{-1} \mathbf{A}\} \mathbf{x}_{t-1} + \\ &\quad \mathbf{x}_{t-1} \{\boldsymbol{v}_t^{-1} \boldsymbol{\mu}_t^T + \frac{1}{2}\mathbf{A}^T \mathbf{Q}^{-1} (\mathbf{x}_t - \mathbf{B}\mathbf{y}_{t-1})\} + \\ &\quad \log(Kx). \end{aligned} \tag{2.111}$$

Collecting the linear and quadratic terms of the state $\mathbf{x}_{t-1}$, we can define a new mean, $\boldsymbol{\mu}_x$, and covariance, $\boldsymbol{\sigma}_x$, for the state $\mathbf{x}_{t-1}$,

Quadratic term: $\mathbf{x}_{t-1}^T \underbrace{\{\boldsymbol{v}_t^{-1} + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}}_{\Lambda} \mathbf{x}_{t-1}$,

Linear term: $\mathbf{x}_{t-1}^T \underbrace{\{\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t^T + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}(\mathbf{x}_t - \mathbf{B}\mathbf{y}_{t-1})\}}_{\eta}$.

Therefore the mean, μ_x and covariance, σ_x, can be written as:

$$\begin{aligned}
\boldsymbol{\Sigma}_x &= \boldsymbol{\Lambda}^{-1}, \\
&= \{\boldsymbol{v}_t^{-1} + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}^{-1}, \qquad (2.112) \\
\boldsymbol{\mu}_x &= \boldsymbol{\Lambda}^{-1}\boldsymbol{\eta}, \\
&= \{\boldsymbol{v}_t^{-1} + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\}^{-1}\{\boldsymbol{v}_t^{-1}\boldsymbol{\mu}_t^T + \frac{1}{2}\mathbf{A}^T\mathbf{Q}^{-1}(\mathbf{x}_t - \mathbf{B}\mathbf{y}_{t-1})\}. \qquad (2.113)
\end{aligned}$$

Therefore $P(\mathbf{x}_{t-1}|\mathbf{x}_t; \mathbf{y}_{1:T}, \boldsymbol{\theta})) \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$.

The pseudocode for the Gibbs sampler (as given in the algorithm 2) iterates between two steps; (1) infer all parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Q}, \mathbf{R}\}$, (2) infer the hidden states $\{\mathbf{x}\}$ using forward filtering and backward sampling.

After collecting the samples from the Gibbs sampler algorithm 2 we follow with convergence diagnosis as described in the Section 2.3.

2.2.5 Learning hyperparameters

Generally the covariance functions that we use contain some free parameters. For example, in the list of conditional distribution given in equations 1.42 to 1.47, we observe that the parameters, $\boldsymbol{\Sigma}_A = diag(\sigma_A)^{-1}$, $\boldsymbol{\Sigma}_B = diag(\sigma_B)^{-1}$, $\boldsymbol{\Sigma}_C = diag(\sigma_C)^{-1}$, $\boldsymbol{\Sigma}_D = diag(\sigma_D)^{-1}$, α, β, γ and δ can be varied. Here α, β, γ and δ are the parameters of parameters known as hyperparameters.

In this section we discuss a method used for the learning of such hyperparam-

Algorithm 2: Gibbs Sampler algorithm for the SSM with feedback.

Input: Randomly initialize parameters $\boldsymbol{\mu}_A$, $\boldsymbol{\Sigma}_A$, $\boldsymbol{\mu}_B$, $\boldsymbol{\Sigma}_B$, $\mathbf{Q}$, μ_C, $\boldsymbol{\Sigma}_C$, μ_D, $\boldsymbol{\Sigma}_D$, $\mathbf{R}$, α, β, γ, δ, and the latent variable $\{\mathbf{x}\}$. Fix the length of MCMC chain N. Here k is dimension of state space and p is dimension of observation sequence.

Output: N number of samples of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$, $\mathbf{R}$ and $\{\mathbf{x}\}$

Step I of infering parameters
For each row $i = 1, \ldots, k$
Sample $\boldsymbol{a}_i|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{B}, \mathbf{Q} \sim N(\tilde{\boldsymbol{\mu}}_{a_i}, \tilde{\boldsymbol{\Sigma}}_{a_i})$ using 2.74 and 2.75
update $\mathbf{A}$
Sample $\boldsymbol{b}_i|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{A}, \mathbf{Q} \sim N(\tilde{\boldsymbol{\mu}}_{b_i}, \tilde{\boldsymbol{\Sigma}}_{b_i})$ using 2.81 and 2.82
update $\mathbf{B}$
Sample $\boldsymbol{q}_{ii}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{A}, \mathbf{B} \sim IG(\tilde{\alpha}, \tilde{\beta})$ using 2.102 and 2.103
update $\mathbf{Q}$
For each row $s = 1, \ldots, p$
Sample $\boldsymbol{c}_s|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{D}, \mathbf{R} \sim N(\tilde{\boldsymbol{\mu}}_{c_i}, \tilde{\boldsymbol{\Sigma}}_{c_i})$ using 2.88 and 2.89
Update $\mathbf{C}$
Sample $\boldsymbol{d}_s|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{C}, \mathbf{R} \sim N(\tilde{\boldsymbol{\mu}}_{d_i}, \tilde{\boldsymbol{\Sigma}}_{d_i})$ using 2.95 and 2.96
Update $\mathbf{D}$
Sample $\boldsymbol{r}_{ss}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{C}, \mathbf{D} \sim IG(\tilde{\gamma}, \tilde{\delta})$ using 2.104 and 2.105
Update $\mathbf{R}$
Step II inference of hidden states
Forward sampling: for $t = 1, \ldots, T$
Sample $\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R} \sim N(\boldsymbol{\mu}_t, \boldsymbol{v}_t)$ using Kalman filtering
Backward sampling: for $t = (T-1), \ldots, 1$
Sample $\mathbf{x}_{t-1}|\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \mathbf{y}_{1:T}, \mathbf{x}_t = N(\boldsymbol{\sigma}_x, \boldsymbol{\mu}_x)$ using 2.112
Update $\{\mathbf{x}\}$.
Repeat steps 2-19 till N number of samples of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$, $\mathbf{R}$ and $\{\mathbf{x}\}$ are collected.

eters. Our goal here is simply to examine the effects of varying the hayperparameters on the inference based on the Gibbs sampling algorithm. To do so we have proposed to add conjugate priors on the hyperparameters that we have obtained from each of the parameter matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$. There are Gamma prior on the variances of parameter matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$. On the shape and scalar parameters of the noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ we have applied Gaussian priors.

$$\sigma_A^2 \quad \sim \prod_{i=1}^{k} Ga(\sigma_A^2|\alpha_A, \beta_A) \tag{2.114}$$

$$\sigma_B^2 \sim \prod_{i=1}^{k} Ga(\sigma_B^2|\alpha_B, \beta_B) \tag{2.115}$$

$$\sigma_C^2 \sim \prod_{i=1}^{p} Ga(\sigma_C^2|\alpha_C, \beta_C) \tag{2.116}$$

$$\sigma_D^2 \sim \prod_{i=1}^{p} Ga(\sigma_D^2|\alpha_D, \beta_D) \tag{2.117}$$

$$\alpha \sim N(0, \sigma_\alpha) \tag{2.118}$$

$$\beta \sim N(0, \sigma_\beta) \tag{2.119}$$

$$\gamma \sim N(0, \sigma_\gamma) \tag{2.120}$$

$$\delta \sim N(0, \sigma_\delta). \tag{2.121}$$

By using conjugate priors on the hyperparameters we can update previously fixed hyperparameters in the following way. So far Σ_A and Σ_B were fixed by assigning an identity matrix of the same dimension as $\mathbf{X}$. Assume that the elements of an identity matrix (Σ_A) can be sampled from a gamma distribution with fixed shape and scalar parameters, i.e. $\alpha_A = 2$ and $\beta_A = 1$ respectively, which we denote as ${\sigma_A^2}^{old}$. Now by adding a perturbation of $\delta\alpha \sim N(0, S)$ (where S is some small value), to the shape and scalar parameters of σ_A^2 as α_A and β_A we obtain new shape and scalar parameters, $\tilde{\alpha}_A$ and $\tilde{\beta}_A$,

$$\tilde{\alpha}_A = \alpha_A + \delta\alpha, \qquad \delta\alpha \sim N(0, S)$$

$$\tilde{\beta}_A = \beta_A + \delta\beta, \qquad \delta\beta \sim N(0, S)$$

and we can draw ${\sigma_A^2}^{new}$ from $Ga(\tilde{\alpha}_A, \tilde{\beta}_A)$. ${\sigma_A^2}^{new}$ will be accepted or rejected using the following Metropolis $-$Hastings algorithm steps,

1. Suppose a candidate value ${\sigma_A^2}^{new}$ is from the proposal density $Ga(\tilde{\alpha}_A, \tilde{\beta}_A)$,

2. Given the candidate value ${\sigma_A^2}^{new}$, the acceptance probability $\pi({\sigma_A^2}^{new}|{\sigma_A^2}^{old})$

can be given as:

$$\pi(\sigma_A^{2\,new}|\sigma_A^{2\,old}) = min\left(\frac{Ga(\tilde{\alpha}_A, \tilde{\beta}_A)}{G(\alpha_A, \beta_A)}, 1\right),$$

3. The probability of $\pi(\sigma_A^{2\,new}|\sigma_A^{2\,old})$ is set as follows:

   - sample randomly a value u from the uniform distribution $U(0,1)$ based on an interval $(0,1)$
   - If $u \geq \pi(\sigma_A^{2\,new}|\sigma_A^{2\,old})$, then candidate value $\sigma_A^{2\,new}$ is accepted and set $\sigma_A^{2\,old} = \sigma_A^{2\,new}$. Otherwise the candidate value $\sigma_A^{2\,new}$ will be rejected and set to $\sigma_A^{2\,new} = \sigma_A^{2\,old}$

4. Repeat steps 1-3 untill the acceptance rate of optimal MH i.e., $(0.2 - 0.4)$ has achieved [Gilks et al., 1996](Chapter 2).

Similarly the inference of the other hyperparameters follows in a similar fashion. The noise covariance hyperparameters will follow the proposal density of $N(0, \sigma_\alpha)$ following the proposal distributions defined in 2.114 to 2.121. The pseudo−code for algorithm 3 introduces the hyperparameter learning step by including the Metropolis Hastings algorithm steps within the Gibbs sampling algorithm.

2.3 Convergence Diagnostics

After collecting a sufficiently large set of samples from the MCMC run, our next step will be to assess the convergence of the parameters towards the stationary distribution. In this section we will review how convergence may be diagnosed, firstly by visualization and secondly through the numerical evaluation. For the visualisation we have mainly used the cumulative average of the drawn samples, and used trace plots to present and compare MCMC chains from different initial values. In addition to trace plots, considering the gradient of the trace plot can also

Algorithm 3: MetropolisHastings with in Gibbs Sampler algorithm from an SSM with feedback loop.

Input: Randomly initialize parameters $\boldsymbol{\mu}_A$, $\boldsymbol{\Sigma}_A$, $\boldsymbol{\mu}_B$, $\boldsymbol{\Sigma}_B$, $\mathbf{Q}$, μ_C, $\boldsymbol{\Sigma}_C$, μ_D, $\boldsymbol{\Sigma}_D$, $\mathbf{R}$, α, β, γ, δ, and the latent variable $\{\mathbf{x}\}$. Fix the length of MCMC chain N. Here k is dimension of state space and p is dimension of observation sequence.

Output: N number of samples of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$, $\mathbf{R}$ and $\{\mathbf{x}\}$

Step Ia of infering parameters
For each row $i = 1, \ldots, k$
Sample $\boldsymbol{a}_i|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{B}, \mathbf{Q} \sim N(\tilde{\boldsymbol{\mu}}_{a_i}, \tilde{\boldsymbol{\Sigma}}_{a_i})$ using 2.74 and 2.75
update $\mathbf{A}$
Sample $\boldsymbol{b}_i|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{A}, \mathbf{Q} \sim N(\tilde{\boldsymbol{\mu}}_{b_i}, \tilde{\boldsymbol{\Sigma}}_{b_i})$ using 2.81 and 2.82
update $\mathbf{B}$
Sample $\boldsymbol{q}_{ii}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{A}, \mathbf{B} \sim IG(\tilde{\alpha}, \tilde{\beta})$ using 2.102 and 2.103
update $\mathbf{Q}$
For each row $s = 1, \ldots, p$
Sample $\boldsymbol{c}_s|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{D}, \mathbf{R} \sim N(\tilde{\boldsymbol{\mu}}_{c_i}, \tilde{\boldsymbol{\Sigma}}_{c_i})$ using 2.88 and 2.89
Update $\mathbf{C}$
Sample $\boldsymbol{d}_s|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{C}, \mathbf{R} \sim N(\tilde{\boldsymbol{\mu}}_{d_i}, \tilde{\boldsymbol{\Sigma}}_{d_i})$ using 2.95 and 2.96
Update $\mathbf{D}$
Sample $\boldsymbol{r}_{ss}|\{\mathbf{x}\}, \{\mathbf{y}\}, \mathbf{C}, \mathbf{D} \sim IG(\tilde{\gamma}, \tilde{\delta})$ using 2.104 and 2.105
Update $\mathbf{R}$
Step Ib of updating hyperparmeters using MetropolisHastings algorithm
Update Σ_A, Σ_B, α, β, Σ_C, Σ_D, γ and δ following steps from section 2.2.5
Step II of infering hidden state
Forward sampling: for $t = 1, \ldots, T$
Sample $\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R} \sim N(\boldsymbol{\mu}_t, \boldsymbol{v}_t)$ using Kalman filtering
Backward sampling: for $t = (T-1), \ldots, 1$
Sample $\mathbf{x}_{t-1}|\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \mathbf{y}_{1:T}, \mathbf{x}_t = N(\boldsymbol{\sigma}_x, \boldsymbol{\mu}_x)$ using 2.112
Update $\{\mathbf{x}\}$.
Repeat steps 2-21 till N number of samples of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$, $\mathbf{R}$ and $\{\mathbf{x}\}$ are collected.

be useful. Visual plots show that during initial iterations the gradient of Markov chain might fluctuate, but eventually it would be expected to reach a steady state, indicating that a stationary distribution has been achieved. For measuring the convergence we have use the Gelman and Rubin [1992] diagnostic approach.

2.3.1 Gelman and Rubin Multiple Sequence Diagnostics

Gelman and Rubin proposed "the potential scale reduction factor" (in short psrf) convergence diagnostic for multiple sequences ($\mathbf{s}$) of draws from a Markov chain. This analysis requires at least 2 chains (that could be as many as $m \geq 2$) of length $2n$, where obviously n is half of the total length, from nonidentical starting values. After discarding the first n draws in each chain we calculate the variance within-chain and between-chain. The variance within-chain can be calculated using

$$W = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{s}_j^2,$$

where

$$\boldsymbol{s}_j^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(\theta_{ij} - \mu_{\theta_j})^2.$$

Here $\boldsymbol{s}_j^2$ defines the variance of the j^{th} chain, θ is the model parameter of interest, μ_{θ_j} defines the mean of the j^{th} chain and W is the average of the variance of m chains. The between-chain variance can be calculated using

$$B = \frac{1}{m-1}\sum_{j=1}^{m}(\mu_{\theta_j} - \hat{\mu_\theta})^2,$$

where, $\hat{\mu_\theta}$ is average of the means of m chains. By using the variance within and between chains we can estimate the variance of the stationary distribution as a weighted average of W and B,

$$var(\theta) = (1 - \frac{1}{n})W + B.$$

The potential scale reduction factor is then defined as

$$R = \sqrt{\frac{var(\theta)}{W}}. \tag{2.122}$$

When the R is greater than any value within the range $1.1 - 1.2$ this indicates that the variance of the stationary distribution is high, which further means that for convergence the chains need to have more iterations. As we have more than one parameter in our model we calculated the potential scale reduction factor for each of the inferred parameters.

2.4 Model Selection: Calculating Marginal Likelihood from the Gibbs Sampler Output

In this section we aim to do model selection for up to K models, M_K. The density function of the data $\{\mathbf{y}\} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t)$ can be written as $P(\mathbf{y}|\theta_k, M_k)$ where M_k, for $k = 1, 2, \cdots, K$, is a model with parameter vector $\boldsymbol{\theta}_k$. Let the prior density of $\boldsymbol{\theta}_k$ be written as $P(\boldsymbol{\theta}_k|M_k)$, and let $\{\boldsymbol{\theta}_k^{(g)}\} = \boldsymbol{\theta}_k^{(1)}, \cdots, \boldsymbol{\theta}_k^{(G)}$ be G draws from the posterior density $P(\boldsymbol{\theta}_k|\mathbf{y}, M_k)$ obtained using the Gibbs sampler. Work by Chib [1995] describes how other methods used to calculate the marginal likelihood have certain limitations. For example, the method of Newton and Raftery shows that the marginal likelihood under model M_k can be estimated as

$$\hat{m}_{NR} = \left\{ \frac{1}{2} \sum_{g=1}^{G} \left(\frac{1}{P(\mathbf{y}|\theta_k^{(g)}, M_k)} \right) \right\}^{-1}. \tag{2.123}$$

The equation 2.123 shows the harmonic mean of the likelihood values. However this approach is not considered to be stable, since the inverse of likelihood does not have finite variance. Gelfand and Smith [Gelfand and Smith, 1990] proposed the use of the quantity

$$\hat{m}_{GD} = \left\{ \frac{1}{G} \sum_{g=1}^{G} \left(\frac{P(\theta_k^{(g)})}{P(\mathbf{y}|\theta_k^{(g)}, M_k) P(\theta_k^{(g)}|M_k)} \right) \right\}^{-1}. \tag{2.124}$$

In equation 2.124, $P(\theta)$ is a density with thinner tails than the product of the likelihood and the prior. This expression has the property that $\hat{m}_{GD} \rightarrow P(\mathbf{y}|M_k)$ as $G \rightarrow \infty$. However, one of the requirements of this approach is to define a tuning function, which is not easy to calculate especially for high dimensional problems.

However, Chib's method [Chib, 1995] demonstrates a simple approach to compute the marginal likelihood and the Bayes factor that is free from the limitations observed in the methods mentioned above. This approach is developed for the case where a Gibbs sampling algorithm has been used to provide draws from the posterior distribution. One of the requirements of this approach is that all normalising constants of the full conditional distributions in the Gibbs sampler are known. However as shown in Section 2.2.1, the Gibbs sampler uses conjugate priors for which the constants can be calculated.

2.4.1 The Chib approach

Consider the situation where $P(\{\mathbf{y}\}|\theta)$ is the likelihood function for the given model and $P(\theta)$ is the prior density. Let $\{\mathbf{x}\}$ be the hidden states and initially suppose that for a set of vector blocks $\theta = (\theta_1, \theta_2, \cdots, \theta_B)$ the Gibbs sampling algorithm is applied to the set of $(B+1)$ complete conditional densities

$$\{P(\theta_r|\{\mathbf{y}\}, \theta_s(s \neq r), \{\mathbf{x}\})\}_{r=1}^{B}, \tag{2.125}$$

$$P(\{\mathbf{x}\}|\{\mathbf{y}\}, \theta). \tag{2.126}$$

The core of this method is to compute the marginal density $m(\mathbf{y}|M_k)$ from the Gibbs output $\{\theta^{(g)}, \{\mathbf{x}\}^{(g)}\}_{g=1}^{G}$ obtained from 2.125, where the index g represents each drawn sample from a total G samples. Chib's method consists of two related ideas. Firstly, by Bayes rule the normalising constant of the posterior density can be written as

$$m(y) = \frac{P(\mathbf{y}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|\mathbf{y})},$$

where the numerator is just the product of sample density (likelihood) and the prior (with all integrating constants included) and the denominator is the posterior density of θ.

Secondly, for a given $\boldsymbol{\theta}$ (say $\boldsymbol{\theta}^*$), the posterior probability $P(\boldsymbol{\theta}^*|\{\mathbf{y}\})$ can be estimated by exploiting the information in the collection of complete conditional densities $\{P(\boldsymbol{\theta}_r|\{\mathbf{y}\}, \boldsymbol{\theta}_s(s \neq r), \{\mathbf{x}\})\}_{r=1}^{B}$. If the posterior density estimated at θ^* is denoted by $\hat{P}(\theta^*|y)$, then the proposed estimate of the marginal density, on the logarithmic scale can be written as:

$$\ln \hat{m}(\mathbf{y}) = \ln P(\{\mathbf{y}\}|\boldsymbol{\theta}^*) + \ln P(\boldsymbol{\theta}^*) - \ln \hat{P}(\boldsymbol{\theta}^*|\{\mathbf{y}\}) \tag{2.127}$$

In the implementation of Chib's method, we first start with a canonical situation that consists of two vector blocks of parameters, and later describe the case with additional vector blocks of parameters.

Case-I

Consider the simple form of a SSM as described by equations 1.1 and 1.2, which has parameters $\theta_A, \theta_C, \theta_Q, \theta_R$ and state vectors $\{\mathbf{x}\}$. We can block parameters θ_A, θ_C together, since they are independent of θ_Q, θ_R and write as two vector blocks i.e. $(B = 2)$

$$\begin{aligned} \theta_1 &= \theta_A, \theta_C|\theta_Q, \theta_R, \{\mathbf{x}\}, \{\mathbf{y}\}, \\ \theta_2 &= \theta_Q, \theta_R|\theta_A, \theta_B, \{\mathbf{x}\}, \{\mathbf{y}\}. \end{aligned}$$

Therefore the $(B+1 = 3)$ complete conditional densities from the Gibbs sampler are

$$P(\theta_1 \mid \theta_2, \{\mathbf{x}\}, \{\mathbf{y}\}),$$

$$P(\theta_2 \mid \theta_1, \{\mathbf{x}\}, \{\mathbf{y}\}),$$

$$P(\mathbf{x} \mid \theta_1, \theta_2, \{\mathbf{y}\}).$$

Suppose that the output of the Gibbs sampler is given by $\{\theta^{(g)}, \{\mathbf{x}\}\}_{g=1}^{G}$, where G is the total number of drawn samples and θ^* is some selected value of the parameter θ. The posterior density can be written as

$$P(\theta_1^* \mid \{\mathbf{y}\}) \cdot P(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\}), \tag{2.128}$$

where

$$P(\theta_1^* \mid \{\mathbf{y}\}) = \int P(\theta_1^* \mid \{\mathbf{y}\}, \theta_2, \{\mathbf{x}\}) P(\theta_2, \{\mathbf{x}\} | \{\mathbf{y}\}) d\theta_2 d\mathbf{x}, \tag{2.129}$$

$$P(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\}) = \int P(\theta_2^* \mid \{\mathbf{y}\}, \theta_1^*, \{\mathbf{x}\}) P(\{\mathbf{x}\} | \{\mathbf{y}\}, \theta_1^*) d\mathbf{x}. \tag{2.130}$$

Equation 2.130 is referred to as the reduced conditional density. Equation 2.129 can be estimated by taking the ergodic average of the full conditional density with the posterior draws of $(\theta_2, \{\mathbf{y}\})$,

$$\hat{P}(\theta_1^* \mid \{\mathbf{y}\}) = \frac{1}{G} \sum_{g=1}^{G} P(\theta_1^* \mid \{\mathbf{y}\}, \theta_2^{(g)}, \{\mathbf{x}\}^{(g)}).$$

The estimate of $\hat{P}(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\})$ can be written as

$$\hat{P}(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\}) = \frac{1}{G} \sum_{g=1}^{G} P(\theta_2^* \mid \{\mathbf{y}\}, \theta_1^*, \{\mathbf{x}\}^{(g)}),$$

where we continue to sample for an additional G iterations from $P(\theta_2^* \mid \{\mathbf{y}\}, \theta_1^*, \{\mathbf{x}\}^{(g)})$ and $P(\{\mathbf{x}\} | \{\mathbf{y}\}, \theta_1^*, \theta_2)$, but with θ_1 set to θ_1^*. Now, substituting $\hat{P}(\theta_1^* \mid \{\mathbf{y}\})$ and $\hat{P}(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\})$ into 2.127 yields the estimate

$$\ln \hat{m}(\mathbf{y}) = \ln P(\{\mathbf{y}\} | \boldsymbol{\theta}^*) + \ln P(\boldsymbol{\theta}^*) - \ln \hat{P}(\theta_1^* \mid \{\mathbf{y}\} - \ln \hat{P}(\theta_2^* \mid \theta_1^*, \{\mathbf{y}\}).$$

Case-II

Now consider the situation with an arbitrary number of blocks B, and suppose the

Gibbs algorithm is defined through the following $(B+1)$ conditional densities as given in equation 2.125. Therefore the $\hat{P}(\theta^*|\{\mathbf{y}\})$ can be expressed as

$$\hat{P}(\theta|\{\mathbf{y}\}) = P(\theta_1^* \mid \{\mathbf{y}\}) \times P(\theta_2^* \mid \{\mathbf{y}\}, \theta_1^*) \times \cdots \times P(\theta_B^* \mid \{\mathbf{y}\}, \theta_1^*, \cdots, \theta_{B-1}^*),$$

The reduced conditional densities needed to estimate $\hat{P}(\theta^* \mid \{\mathbf{y}\})$ can generally be given as

$$\begin{aligned} P(\theta_r^*|\{\mathbf{y}\}, \theta_1^*, \theta_2^*, \cdots, \theta_{r-1}^*) &= \int P(\theta_r^*|\{\mathbf{y}\}, \theta_1^*, \theta_2^*, \cdots, \theta_{r-1}^*, \theta_{r+1}^*, \cdots, \theta_B, \{\mathbf{x}\}) \\ &\quad dP(\theta_{r+1}, \cdots, \theta_B, \{\mathbf{x}\}|\{\mathbf{y}\}, \theta_1^*, \theta_2^*, \cdots, \theta_{r-1}^*), \end{aligned}$$

so the estimate of

$$P(\theta_r^* \mid \{\mathbf{y}\}, \theta_1^*, \cdots, \theta_{r-1}^*)$$

can be given by taking the ergodic average

$$\hat{P}(\theta_r^* \mid \{\mathbf{y}\}, \theta_1^*, \cdots, \theta_{r-1}^*) = \frac{1}{G} \sum_{g=1}^{G} P(\theta_r^* \mid \{\mathbf{y}\}, \theta_1^*, \cdots, \theta_{r-1}^*, \theta_{r+1}^{(g)}, \cdots, \theta_B^{(g)}, \{\mathbf{x}\}^{(g)}).$$

The log marginal likelihood is then written as

$$\ln \hat{m}(y) = \ln P(\{\mathbf{y}\}|\theta^*) + \ln P(\theta^*) - \sum_{r=1}^{B} \ln \hat{P}(\theta_r^* \mid \{\mathbf{y}\}, \theta_1^*, \cdots, \theta_{r-1}^*).$$

In the SSM with feedback case discussed in this thesis we set the vector blocks for parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{A,B}, \boldsymbol{\theta}_Q, \boldsymbol{\theta}_{C,D}, \boldsymbol{\theta}_R\}$ i.e. a vector block of size $B = 4$. Therefore, in this case, the Gibbs sampling algorithm is applied to a set of $(B+1) = 5$ complete conditional densities,

$$P(\boldsymbol{\theta}_{A,B}|\{\mathbf{y}\}, \boldsymbol{\theta}_Q, \{\mathbf{x}\}),$$

$$P(\boldsymbol{\theta}_Q|\{\mathbf{y}\}, \boldsymbol{\theta}_Q, \boldsymbol{\theta}_{A,B}, \{\mathbf{x}\}),$$

$$P(\boldsymbol{\theta}_{C,D}|\{\mathbf{y}\}, \boldsymbol{\theta}_R, \{\mathbf{x}\}),$$

$$P(\boldsymbol{\theta}_R|\{\mathbf{y}\}, \boldsymbol{\theta}_{C,D}, \{\mathbf{x}\}),$$
$$P(\{\mathbf{x}\}|\{\mathbf{y}\}, \boldsymbol{\theta}). \tag{2.131}$$

Now, suppose from the total number of drawn samples G the output of the Gibbs sampler is given by $\{\theta^{(g)}_{A,B}, \theta^{(g)}_{Q}, \theta^{(g)}_{C,D}, \theta^{(g)}_{R}, \{\mathbf{x}\}^{(g)}\}$ and θ^* is some selected point[2]. Hence the posterior density can be written as

$$P(\boldsymbol{\theta}^*_{A,B}|\boldsymbol{\theta}_Q, \{\mathbf{x}\}, \{\mathbf{y}\})P(\boldsymbol{\theta}^*_{Q}|\boldsymbol{\theta}^*_{A,B}, \{\mathbf{x}\}, \{\mathbf{y}\})P(\boldsymbol{\theta}^*_{C,D}|\boldsymbol{\theta}_R, \{\mathbf{x}\}, \{\mathbf{y}\}),$$
$$P(\boldsymbol{\theta}^*_{R}|\boldsymbol{\theta}^*_{C,D}, \{\mathbf{x}\}, \{\mathbf{y}\}). \tag{2.132}$$

Where each term of equation 2.132 comes from the following integrations.

$$P(\boldsymbol{\theta}^*_{A,B}|\{\mathbf{y}\}) = \int P(\boldsymbol{\theta}^*_{A,B}|\boldsymbol{\theta}_Q, \{\mathbf{x}\}, \{\mathbf{y}\})\, dP(\boldsymbol{\theta}_Q, \{\mathbf{x}\}|\{\mathbf{y}\}) \tag{2.133}$$

$$P(\boldsymbol{\theta}^*_{Q}|\{\mathbf{y}\}) = \int P(\boldsymbol{\theta}^*_{Q}|\boldsymbol{\theta}^*_{A,B}, \{\mathbf{x}\}, \{\mathbf{y}\})\, dP(\{\mathbf{x}\}|\boldsymbol{\theta}^*_{A,B}, \{\mathbf{y}\}) \tag{2.134}$$

$$P(\boldsymbol{\theta}^*_{C,D}|\{\mathbf{y}\}) = \int P(\boldsymbol{\theta}^*_{C,D}|\boldsymbol{\theta}_R, \{\mathbf{x}\}, \{\mathbf{y}\})\, dP(\boldsymbol{\theta}_R, \{\mathbf{x}\}|\{\mathbf{y}\}) \tag{2.135}$$

$$P(\boldsymbol{\theta}^*_{R}|\{\mathbf{y}\}) = \int P(\boldsymbol{\theta}^*_{R}|\boldsymbol{\theta}^*_{C,D}, \{\mathbf{x}\}, \{\mathbf{y}\})\, dP(\{\mathbf{x}\}|\boldsymbol{\theta}^*_{C,D}, \{\mathbf{y}\}) \tag{2.136}$$

which are the reduced conditional densities. Equation 2.133 can be estimated by taking the ergodic average of the full conditional density with the posterior draws of $(\boldsymbol{\theta}^{(g)}_{Q}, \{\mathbf{x}\}^{(g)})^G_{g=1}$

$$\hat{P}(\boldsymbol{\theta}^*_{A,B}|\mathbf{y}) = G^{-1}\sum_{g=1}^{G} P(\boldsymbol{\theta}^*_{A,B}|\boldsymbol{\theta}^{(g)}_{Q}, \mathbf{x}^{(g)}, \mathbf{y})$$

The reduced conditional probability $P(\boldsymbol{\theta}^*_{Q}|\boldsymbol{\theta}^*_{A,B}, \mathbf{y})$ can be estimated by taking the

[2]The choice of the point is not critical. It is usually chosen to be a high density point where large number of samples are available. A modal value such as the posterior mode or the maximum likelihood estimate, which can be approximately computed from the Gibbs sampler output or the posterior mean is suitable, provided that it is not picked from a low density point (such as the tail region).

average of the conditional density with $\theta_{A,B}$ set to be $\boldsymbol{\theta}^*_{A,B}$ for another G iterations. This leads to the estimate

$$\hat{P}(\boldsymbol{\theta}^*_Q|\mathbf{y}, \boldsymbol{\theta}^*_{A,B}) = G^{-1} \sum_{g=1}^{G} P(\boldsymbol{\theta}^*_Q|\mathbf{y}, \boldsymbol{\theta}^*_{A,B}, \mathbf{x}^{(g)}).$$

Similarly for the probability $P(\boldsymbol{\theta}^*_{C,D}|\mathbf{y})$ the estimate can be given as

$$\hat{P}(\boldsymbol{\theta}^*_{C,D}|\mathbf{y}) = G^{-1} \sum_{g=1}^{G} P(\boldsymbol{\theta}^*_{C,D}|\boldsymbol{\theta}^{(g)}_R, \mathbf{x}^{(g)}, \mathbf{y}),$$

and the estimate for the probability $P(\boldsymbol{\theta}^*_R|\mathbf{y}, \boldsymbol{\theta}^*_{C,D})$ can be calculated as

$$\hat{P}(\boldsymbol{\theta}^*_R|\mathbf{y}, \boldsymbol{\theta}^*_{C,D}) = G^{-1} \sum_{g=1}^{G} P(\boldsymbol{\theta}^*_R|\mathbf{y}, \boldsymbol{\theta}^*_{C,D}, \mathbf{x}^{(g)}).$$

Now, substituting the above four densities estimate into 2.127 we get

$$\begin{aligned} \ln \hat{m}(\mathbf{y}) \quad = \quad & \ln P(\mathbf{y}|\boldsymbol{\theta}^*) + \ln P(\boldsymbol{\theta}^*) - \ln \hat{P}(\boldsymbol{\theta}^*_{A,B}|\mathbf{y}) - \ln \hat{P}(\boldsymbol{\theta}^*_Q|\mathbf{y}, \boldsymbol{\theta}^*_{A,B}) \\ & - \ln \hat{P}(\boldsymbol{\theta}^*_{C,D}|\mathbf{y}) - \ln \hat{P}(\boldsymbol{\theta}^*_R|\mathbf{y}, \boldsymbol{\theta}^*_{C,D}). \end{aligned} \tag{2.137}$$

2.5 Summary

In this chapter we describe an algorithm based on the Gibbs sampler for SSMs. Initially we have shown the implementation of an MCMC algorithm based on the canonical form of state space model. However later we extend this to a SSM with feedback and demonstrate the derivation of full conditional distributions required for the Gibbs Sampler. The fixed hyperparamters were varied using a Metropolis−Hastings step within the Gibbs Sampler. After building an algorithm we demonstrate the calculation of marginal likelihood from the Gibbs output. The calculated marginal likelihood will be then used for the purpose of model selection.

As mentioned earlier this chapter gives the methodology used to build a

MCMC sampler based algorithm. In Chapter 3 we will show the validation of the proposed algorithm and learning of hyperparameters. Chapter 4 will demonstrate reverse engineering of an *in silico* network. These two chapters evaluate the performance of the Gibbs sampler to ensure that the proposed algorithm is reliable before application to the reverse engineering task using more realistic datasets.

Chapter 3

Application to simulated data

3.1 Introduction

In Chapter 2 a Gibbs sampler for a state space model (SSM) with feedback was described. This chapter describes and demonstrates numerical experiments to validate the Gibbs sampler. At a molecular level, the interpretation of experimental observations (data) and the understanding of biological processes requires a lot of careful statistical analysis. The complexity of the biological system itself and the considerable amount of quantitative analysis makes modelling the behaviour of such systems challenging. Therefore before applying the proposed algorithm on real world data we take a crucial step of validating its performance.

This chapter has been divided into three sections. Each section contains an explanation of the experiment and supporting results. Section 3.2 describes the first experiment to validate the Gibbs sampler. In this section we have defined a test algorithm as given in Algorithm 4. Following this test it could be confirmed that the parameters converge to their prior distribution, given randomly generated observations, $\{\mathbf{y}\}$. Section 3.3 has three subsections; the first part describes how simulated data has been generated using a SSM of gene expression as described by Rangel [2003], the second part describes the evaluation of our algorithm using simulated

data to recover the parameters of the generating model; and the third part demonstrates the learning of hyperparameters by introducing Metropolis-Hastings steps within the Gibbs sampler. Section 3.5 provides summary of the present chapter.

3.2 Validation algorithm for the Gibbs sampler

The procedure for validating the Gibbs sampler is given in Algorithm 4. We assume the dimension of the state space, $k = 2$, an observation vector of dimension $p = 4$ and time points $T = 100$ to be known. We randomly initialize the parameter vector $\Theta = \{\mathbf{A},\ \mathbf{B},\ \mathbf{Q},\ \mathbf{C},\ \mathbf{D},\ \mathbf{R}\}$. It is common practice in MCMC inference that we are given an observation sequence, $\{\mathbf{y}\}$, and the parameters and the hidden states $(\Theta, \{\mathbf{x}\})$ are inferred. But for the purpose of validation we will sample from all variables using MCMC and will check if the posterior distributions of the parameters converge to their prior distributions i.e. $\Theta \sim P(\Theta)$.

Algorithm 4: Algorithm to test the MCMC method for a SSM.

Input: Randomly initialize parameters $\Theta \equiv \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{Q}, \mathbf{R}$, latent variables $\{\boldsymbol{x}\}$ and observed variables $\{\boldsymbol{y}\}$.

1. Sample $\Theta|\{\boldsymbol{x}\}, \{\boldsymbol{y}\}$; follows from usual MCMC inference.
2. Sample $\{\boldsymbol{x}\}|\Theta, \{\boldsymbol{y}\}$; follows from usual MCMC inference.
3. Sample $\{\boldsymbol{y}\}|\Theta, \{\boldsymbol{x}\}$; random data generated using parameters and states from (2) and (3).
4. Go to (2) until convergence.

Output: $\Theta \sim P(\Theta)$

In the first step of the test algorithm outlined in Algorithm (4), the parameters, Θ, latent variables $\{\mathbf{x}\}$, and observations, $\{\mathbf{y}\}$, have been initialized randomly. Each row of parameter matrix was assigned a conjugate prior. For the parameters $\mathbf{A}, \mathbf{C}$ and $\mathbf{Q}$ with the row index $i = 1, \ldots, k$,

$$P(\boldsymbol{A}_i) = N(\boldsymbol{\mu}_{A_i}, \boldsymbol{\Sigma}_{A_i}),\ P(\boldsymbol{C}_i) = N(\boldsymbol{\mu}_{C_i}, \boldsymbol{\Sigma}_{C_i}),\ P(\boldsymbol{Q}_i) = IG(\alpha, \beta).$$

For the parameters $\mathbf{B}, \mathbf{D}$ and $\mathbf{R}$ with the row index $s = 1, \ldots, p$,

$$P(\boldsymbol{B}_s) = N(\boldsymbol{\mu}_{B_s}, \boldsymbol{\Sigma}_{B_s}),\ P(\boldsymbol{D}_s) = N(\boldsymbol{\mu}_{D_s}, \boldsymbol{\Sigma}_{D_s}),\ P(\boldsymbol{R}_s) = IG(\gamma, \delta).$$

Any $\boldsymbol{\Sigma}'$s stated above are a diagonal covariance matrices and IG stands for the inverse gamma distribution.

For the 1^{st} row of each of the parameter matrices from Θ, the mean and the covariance matrix of the multivariate normal prior distributions are shown in the first two columns of Table 3.1. The set of parameters $\boldsymbol{A}_{1j}, \boldsymbol{C}_{1j}, \boldsymbol{Q}_{1j}$ are for $j = 1, 2$, and $\boldsymbol{B}_{1,s}, \boldsymbol{D}_{1,s}, \boldsymbol{R}_{1,s}$ are for $s = 1, \ldots, 4$. The distributions of the selected parameters are shown in Figure 3.1. In steps 2 and 3 of the test algorithm, we sample the parameters and state sequence, given the observation sequence as usual using the initial values of the parameters, Θ, latent variables $\{\mathbf{x}\}$, and observation variables $\{\mathbf{y}\}$, as given in step 1. In step 4 we generate random data from the current sampled states and parameters using an optional step. This will iterate over steps (2) to (4) until the algorithm achieves convergence.

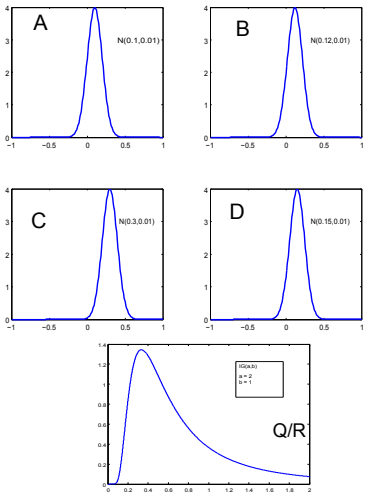


Figure 3.1: The marginal prior distributions set for $\boldsymbol{a}_{11}$,$\boldsymbol{b}_{11}$,$\boldsymbol{c}_{11}$,$\boldsymbol{d}_{11}$ and combined $\boldsymbol{q}_{11}$ *and* $\boldsymbol{r}_{11}$ of the model parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$. These parameters were drawn using the mean and variance specified in the figure legends. For the inverse gamma function we have used the shape (a) and scalar (b) parameters of 2 and 1 respectively.

After collecting a sufficiently large number of samples, the next step is the analysis of convergence. Convergence can be monitored by visual inspection of the cumulative average of a number of drawn samples. Figure 3.2 represents the cumulative average of drawn samples from five different MCMC chains for the first element of model parameters Θ i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11}, r_{11}. Here we observe that despite having different starting points these chains have been properly mixed. Additional plots for all the other elements of the parameter matrices **A**, **B**, **C**, **D**, **Q** and **R** are given in the CD as supplementary material.

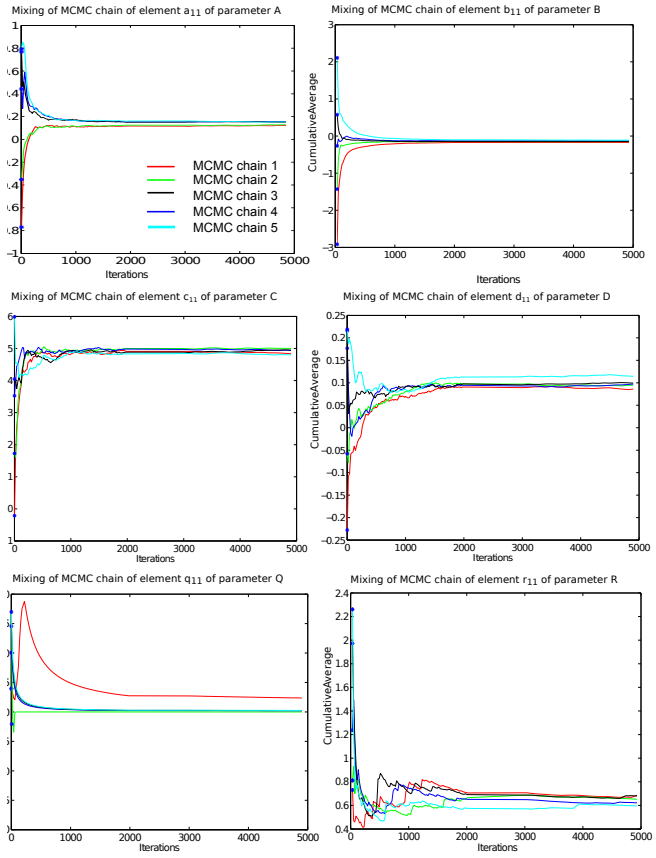


Figure 3.2: The cumulative average of drawn samples for a selection of parameters. These plots demonstrate the convergence of different MCMC chains. The chosen parameters are the first elements of parameters i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11}, r_{11}. All the subplots in this figure share the same legend as mentioned in the first subplot. Despite of reaching stationary distribution the first MCMC chain of an element q_{11} doesnot shares the same parameter space as other chains. Therefore such a chain was excluded for further estimation of the posterior mean.

Para.	Prior mean (μ) Prior variance (Σ)	Posterior Mean (95% CI) Estimated variance	KL div
$\boldsymbol{A_{1,j}}$	$\boldsymbol{\mu_{A_{1,j}}} = [0.1 \;\; 0.1]$	$[-0.0574 \;\; 0.2655]$	0.0242
		$[-0.0630 \;\; 0.2657]$	0.0248
	$\boldsymbol{\Sigma_{A_{1,j}}} = 0.01 \times I_{2\times 2}$	diag($[0.0096 \;\; 0.010]$)	
$\boldsymbol{B_{1,s}}$	$\boldsymbol{\mu_{B_{1,j}}} = [0.12 \;\; 0.12$ $0.12 \;\; 0.12]$	$[-0.0621 \;\; 0.2398]$	0.0265
		$[-0.0629 \;\; 0.2392]$	0.0260
		$[-0.0621 \;\; 0.2388]$	0.0264
		$[-0.0628 \;\; 0.2419]$	0.0261
	$\boldsymbol{\Sigma_{B_{1,j}}} = 0.01 \times I_{4\times 4}$	diag($[0.0084 \;\; 0.0084$ $0.009 \;\; 0.0086]$)	
$\boldsymbol{C_{1,j}}$	$\boldsymbol{\mu_{C_{1,j}}} = [0.5 \;\; 0.5]$	$[0.1353 \;\; 0.4655]$	0.0599
		$[0.1342 \;\; 0.4627]$	0.0604
	$\boldsymbol{\Sigma_{C_{1,j}}} = 0.01 \times I_{2\times 2}$	diag($[0.0101 \;\; 0.01]$)	
$\boldsymbol{D_{1,s}}$	$\boldsymbol{\mu_{D_{1,j}}} = [0.15 \;\; 0.15$ $0.15 \;\; 0.15]$	$[-0.0143 \;\; 0.3124]$	0.0187
		$[-0.0160 \;\; 0.3145]$	0.0186
		$[-0.0161 \;\; 0.3192]$	0.0188
		$[-0.0140 \;\; 0.3145]$	0.0182
	$\boldsymbol{\Sigma_{D_{1,j}}} = 0.01 \times I_{4\times 4}$	diag($[0.0099 \;\; 0.0101$ $0.0104 \;\; 0.100]$)	
Para.	Initial mean (μ)	Est. Mean(EM) [EM $\pm$95% conf. int.] (true value)	
$\boldsymbol{Q_{1,j}}$	$\alpha = 2 \;\; \beta = 1$	$[-0.8994 \;\; 2.2230]$	0.0560
		$[-0.7107 \;\; 2.0609]$	0.0539
	$\boldsymbol{\mu_{Q_{1,j}}} = I_{2\times 2}$		
$\boldsymbol{R_{1,s}}$	$\alpha = 2 \;\; \beta = 1$	$[-0.2124 \;\; 1.1330]$	0.0729
		$[-0.5979 \;\; 1.9487]$	0.0772
		$[-0.1950 \;\; 1.1718]$	0.0857
		$[-0.1515 \;\; 1.1130]$	0.0696
	$\boldsymbol{\mu_{R_{1,j}}} = I_{4\times 4}$		

Table 3.1: Initialization of prior parameters and estimation of mean and covariance for the 1^{st} row of parameter matrices **A**, **B**, **C**, **D**, **Q** and **R**. Last column shows the Kullback-Leibler divergence calculated between prior and posterior distribution. Smallar value of the divergence indicates the closeness between the two distributions.

For measuring convergence we have also used the potential scale reduction factor (PSRF) proposed by Brooks and Gelman [1998]. In this method more than one MCMC chain starting from different initial points have been compared. Details of the calculation to evaluate the variance between and within chains is given in Chapter 2 (Section 2.3.1). As we observe from Figure 3.2 the chains appeared to have converged. Table 3.2 measures convergence by showing the PSRF value for the first row of each parameter matrix. The evaluated PSRF values are all below the range of $1.1 - 1.2$ and therefore confirm the convergence.

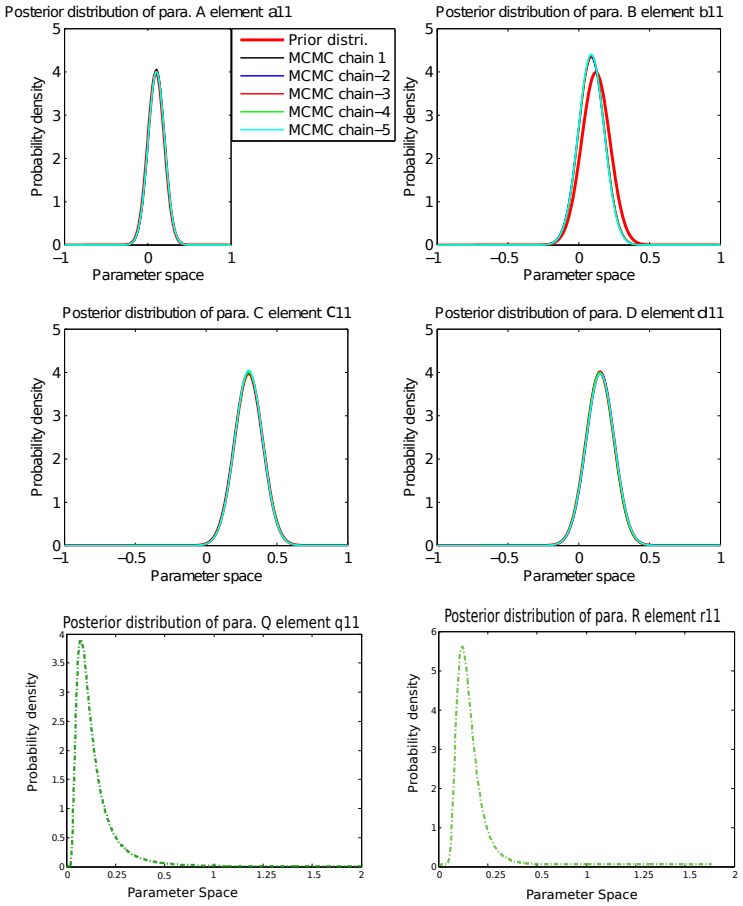


Figure 3.3: The marginal posterior distributions of the first elements of all the parameters i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11}, r_{11}. These distributions are expected to be similar to the prior distributions given in Figure 3.1. All the subplots in this figure shares the same legend as mentioned in the first subplot.

PSRF	$\boldsymbol{A}_{1,:}$	$\boldsymbol{B}_{1,:}$	$\boldsymbol{C}_{1,:}$	$\boldsymbol{D}_{1,:}$	$\boldsymbol{Q}_{1,:}$	$\boldsymbol{R}_{1,:}$
	0.9996	0.9996	0.9983	1.0016	1.0011	1.0222
	0.9992	1.0015	0.9980	0.9996	1.0097	1.0003
		1.0007		1.000		0.9984
		1.0015		1.0006		1.003

Table 3.2: PSRF for the 1^{st} row of parameter matrices **A**, **B**, **C**, **D**, **Q** and **R**.

As the Markov chains are converged, our next step is to estimate the mean and variance of the parameters. For this it is usual to consider the stationary distribution of the Markov chain. Here the stationary part was defined by considering all the last 3000 stable samples from the Markov chain. The estimated mean and variance was then calculated using the stationary part of the chain for each of the parameters. Using these estimates we plot the marginal posterior distributions, as shown in Figure 3.3. These plots show the first element of parameters **A**, **B**, **C** and **D** i.e. a_{11}, b_{11}, c_{11}, d_{11}. Different colors are used to represent the density plots of the posterior distribution drawn from different Markov chains.

It is also possible now to observe the prior means of the parameters lie within the 95% confidence interval of the posterior means. For example, as observed in the first and second column of Table 3.1 for the parameter $\boldsymbol{A_{1,:}}$ the mean was set to be $\boldsymbol{\mu}_{\boldsymbol{A_{1,:}}} = [0.1 \quad 0.1]$ and the covariance was set as $\boldsymbol{\Sigma}_{\boldsymbol{A_{1,:}}} = 0.01 \times I_{2\times 2}$. From 3000 stable samples we estimate the mean $\hat{\boldsymbol{A}}_{1,:} = [a_{11} \quad a_{12}]$ to be $[0.104 \quad 0.101]$ and observe that element $a_{11} = 0.104$ lies within 95% confidence interval (CI) of $[-0.0574 \quad 0.2655]$ and $a_{12} = 0.101$ lies within the 95% CI of $[-0.0630 \quad 0.2657]$. The diagonal element of the estimated covariance matrix $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{A_{1,j}}} \approx 0.009 \times I_{2\times 2}$ is close to the prior.

The closeness or similarity of two distributions may be measured by evaluating the Kullback-Leibler(KL) divergence. In our case we would like to find out the similarity between the prior distribution $P(\Theta)$ and the posterior distribution $P(\Theta|\{\mathbf{x}\},\{\mathbf{y}\})$ i.e. $D_{KL}(P(\boldsymbol{A}|.)||P(\boldsymbol{A}))$. The KL divergence between $P(\boldsymbol{A})$ and $P(\boldsymbol{A}|.)$ also denoted as $D_{KL}(P(\boldsymbol{A}|.)||P(\boldsymbol{A}))$ is given in the last column of Table

3.1. For example the KL divergence calculalted between the prior and the posterior distibution of a_{11} i.e. $D_{KL}(P(a_{11}|.)||P(a_{11})) = 0.0242$ and for a_{11} the KL divergene is $D_{KL}(P(a_{12}|.)||P(a_{12})) = 0.0248$. The low KL values (close to 0) signifies that the two distributions are approaching an identical distribution.

Figure 3.3 also shows that the estimated mean from the marginal posterior of q_{11} is 0.99, which is close to the prior mean 1. The estimated mean from the posterior of R_{11} is 1.09 which is close to the prior mean i.e. 1. The posterior distributions of both error covariances have a long tail as expected for an inverse gamma distribution. The KL divergence calculated for the elements of matrices **Q** and **R** is reported in the last column of Table 3.1. We have observed that the posterior distributions shown in Figure 3.3 converge to their prior distributions as shown in Figure 3.1 as expected. Posterior density plots for other elements of the parameters are shown in the supplementary material (on CD).

3.3 Experiment using simulated data to recover the parameters of the generating model

In this section we will begin with generating simulated data using a state space model. Section 3.3.1 mainly follows the approach of Chapter 2 of Rangel [2003] in order to achieve stable time series observations from the SSM. Using simulated data the numerical experiment was carried out to test the performance of the Gibbs sampler. Our overall goal in this section is to estimate the parameters of the generating model by using the proposed Gibbs sampler algorithm.

3.3.1 Generating simulated data

As in the previous Section 3.2 we assume that the dimension of the state and the observation sequences are $k = 2$ and $p = 4$ respectively. The parameters {**A**, **B**, **C**, **D**} are considered to be multivariate Gaussian and the noise covariance parameter

$\{\mathbf{Q}, \mathbf{R}\}$ to be an inverse gamma distribution. For time points $T = 100$ we simulate data $\{\mathbf{y}\}$ by using the state space model given by equations 2.1 and 2.2.

In a SSM any randomly drawn parameter values might not result in stable time series observations. In order to achieve stable observations one should check the stability property of the SSMs. Detailed derivation of relevant properties is given in Chapter 2 of Rangel [2003]. The stability property is known to satisfy the condition shown in equation 3.1,

$$\rho\left(\det\left(\mathbf{M}\right)\right) = \mathbf{\Lambda} < 1, \tag{3.1}$$

where, $\rho(\cdot)$ referes to the spectral radius as defined in the equation 3.1,

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{CA} & \mathbf{CB} + \mathbf{D} \end{bmatrix}.$$

Considering the distribution of parameters defined in Tables 3.1 we can draw a sample to specify the parameter matrices. We used the Matlab function for the random multivariate normal distribution (i.e. *mvnrnd*) to sample parameter matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. The covariance matrices of the Gaussian white noises that were added to the simulated data are $\mathbf{Q}$ and $\mathbf{R}$. The diagonal matrices $\mathbf{Q}$ and $\mathbf{R}$ were sampled using a random inverse gamma distribution (i.e *gamrnd*) of shape parameter ($\alpha = 2$) and scalar parameter ($\beta = 1$). By using these particular shape and scalar parametere we aim to keep the noise variance to be around 1. The following parameter values were thus generated:

$$\mathbf{A} = \begin{pmatrix} 0.3173 & 0.1572 \\ 0.1917 & 0.0453 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0.2015 & 0.0121 & 0.0620 & -0.067 \\ 0.0595 & 0.1225 & 0.1461 & 0.1943 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 0.2838 & 0,3436 \\ 0.3037 & 0.2815 \\ 0.3890 & 0.2544 \\ 0.2281 & 0.2780 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0.2361 & 0.2564 & -0.0227 & 0.2216 \\ 0.0685 & 0.2490 & -0.0282 & 0.1456 \\ -0.0404 & 0.1108 & -0.0393 & 0.3257 \\ 0.2075 & 0.1670 & -0.1004 & 0.2703 \end{pmatrix}$$

$$\mathbf{Q} = \text{diag}([0.96 \;\; 0.85]) \; \mathbf{R} = \text{diag}([0.5398 \;\; 0.5398 \;\; 0.5398 \;\; 0.5398]).$$

The generated parameter values were then used in equation 3.1 resulting the following set of eigenvalues of matrix $\mathbf{M}$ which are all less than 1.

$$\rho\left(\det\left(\mathbf{M}\right)\right) = \begin{pmatrix} 0.9131 \\ 0.2944 \\ -0.1079 \\ -0.0381 \\ 0.1725 \\ 0.1390 \end{pmatrix} < 1.$$

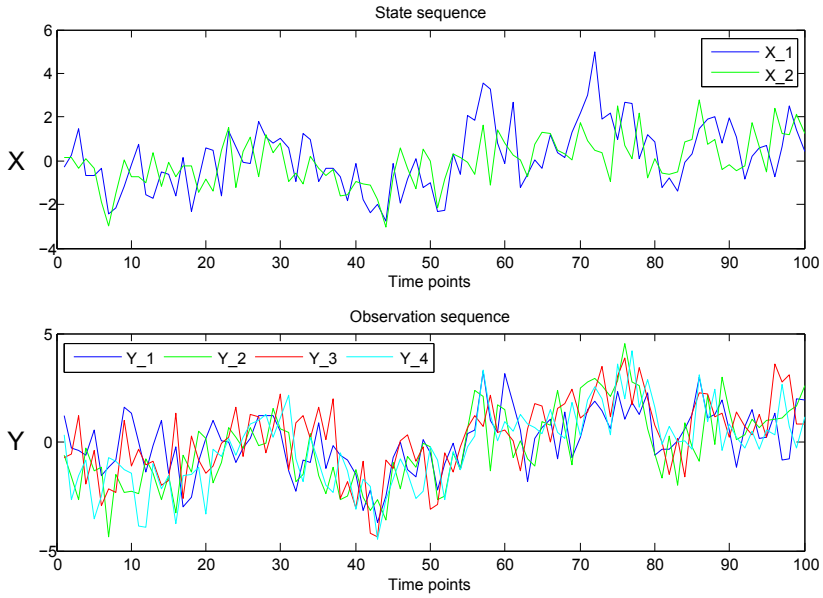


Figure 3.4: Visualisation of simulated state and observation sequences. Results by using model parameters value as defined earlier in this section with spectral radius $\rho < 1$

Therefore the above chosen set of parameter values is suitable to achieve a stable simulated time series, as shown in Figure 3.4. The top plot shows the state sequence of dimension (2×100) and the bottom plot shows the simulated observation sequence of dimension (4×100) generated from an SSM with these parameters.

3.3.2 Numerical Experiment

We consider a simulated observation sequence $\{\mathbf{y}\}$ (see Section 3.3.1) as the observation data, with no other information about the values of parameters. Our aim here is to regain the parameters of generating model as defined in Section 3.2 in Table 3.1 by using the Gibbs sampler. The initialisation of the precisions of the parameters and the state sequence is chosen to be a random multivariate Gaussian variable with non-zero mean and unit covariance. The MCMC algorithm is set to run for $150,000$ iterations from 5 different starting points. In this experiment the initialisation is set far from the parameters of the generating model.

After collecting all drawn samples it is possible now to test convergence. For convergence diagnostics it is advisable to visualize and measure the convergence for different independent MCMC chains (i.e. 5 Markov chains starting from different starting points). In this section we will show the mixing of chains and the behaviour of samples for the first element of parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$ i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11} and r_{11} as given in Figures 3.5 and 3.6. The plots here show the cumulative average of drawn samples. The plots of other elements of parameters are included in the supplementary material.

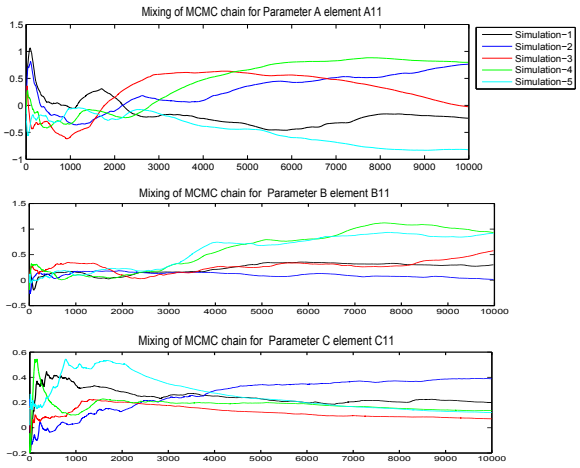


Figure 3.5: The convergence of parameters a_{11}, b_{11} and c_{11} towards their true value.

Figures 3.7 shows the marginal posterior distributions as density plots from

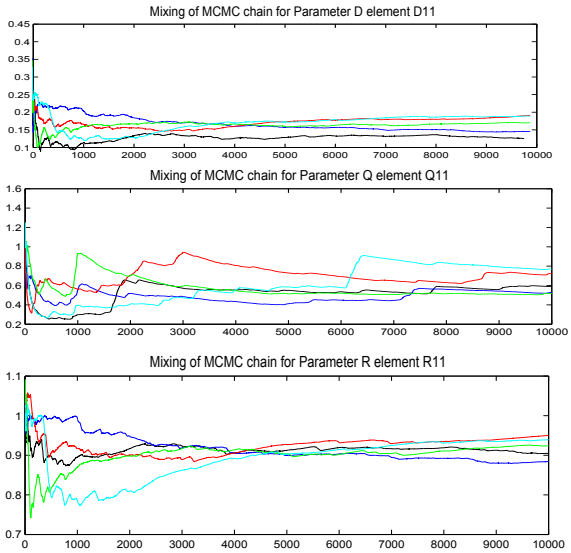


Figure 3.6: The convergence of parameters d_{11}, q_{11}, and r_{11} towards their true value.

different MCMC chains. The plots for other elements of parameter matrices are given on the supplementary CD. Our aim here is to observe how the samples converge to their true distribution from a random initialisation. However as shown in Figure 3.7 for the parameters **A**, **B** and **Q** the convergence of chains has not reached a stationary distribution. Therefore such results would not be suitable for estimating the posterior distribution of parameters. However we believe this may be because of the fixed hyperparameters i.e. α, β, γ, δ of the noise parameters **Q** and **R** in this experiment. We would like to infer the hyperparameters of the noise covariance and also those of the dynamics parameters. To address this issue of updating the hyperparameters, in the next section we introduce a Metropolis-Hasting step within the Gibbs sampler algorithm.

3.3.3 Experiment using Metropolis-Hasting (MH) within Gibbs to retrieve the true parameters

This section demonstrates the implementation of M-H within Gibbs algorithm for a simulated dataset. As with the experiment previously described in Section 3.2,

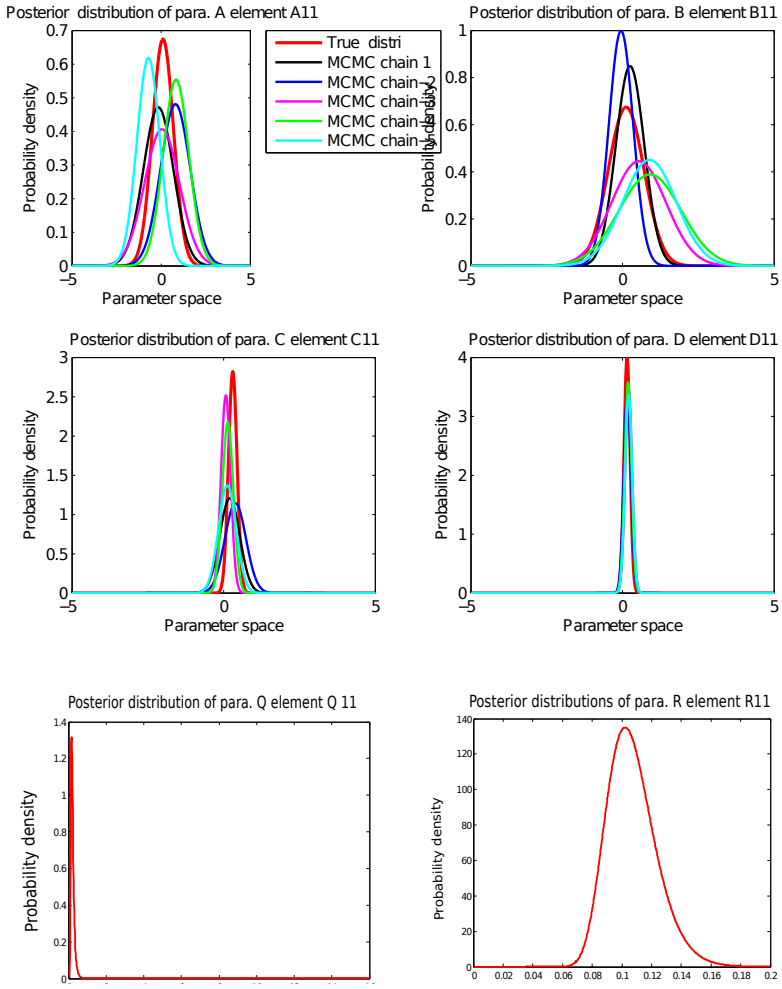


Figure 3.7: The posterior distributions from different MCMC chains, we can see poor mixing for different MCMC chains for the first elements of parameters **A**, **B**, **C**, **D**, **Q** and **R**.

our aim here is to regain the true parameter distributions of the parameters of the model which generated the simulated data. The initialisation of the parameters is defined in the first two columns of Table 3.3.

In this experiment the sampler is set up for a Markov chain of length 50000, for five independent runs. This section demonstrates the convergence towards the true distribution of the parameters. In each plot different colors represent different Markov chains. It can be observed that initially the plots are not smooth but they

Para.	mean (μ) variance (Σ) of generating model	Est. Mean(EM) [EM 95% conf. int.] (true value) Est. Covariance	KL div
$\boldsymbol{A_{1,j}}$	$\boldsymbol{\mu_{A_{1,j}}} = [0.1 \;\; 0.1]$	$[-0.0539 \;\; 0.3703]$ (0.3173)	0.02415
		$[-0.0559 \;\; 0.2682]$ (0.1572)	0.02422
	$\boldsymbol{\Sigma_{A_{1,j}}} = 0.01 \times I_{k\times k}$	$\Sigma_{A_j} = diag([0.00986 \;\; 0.00985])$	
$\boldsymbol{B_{1,j}}$	$\boldsymbol{\mu_{B_{1,j}}} = [0.12 \;\; 0.12$	$[-0.0498 \;\; 0.2499]$ (0.2015)	0.02381
		$[-0.0501 \;\; 0.2505]$ (0.0121)	0.02379
	$0.12 \;\; 0.12]$	$[-0.0505 \;\; 0.2501]$ (0.0620)	0.02398
		$[-0.0503 \;\; 0.2511]$ (-0.0676)	0.02399
	$\boldsymbol{\Sigma_{B_{1,j}}} = 0.01 \times I_{p\times p}$	$\Sigma_{B_j} = diag([0.00911 \;\; 0.00914])$	
$\boldsymbol{C_{1,j}}$	$\boldsymbol{\mu_{C_{1,j}}} = [0.5 \;\; 0.5]$	$[0.1354 \;\; 0.4643]$ (0.2838)	0.0597
		$[0.1367 \;\; 0.4649]$ (0.3436)	0.0589
	$\boldsymbol{\Sigma_{C_{1,j}}} = 0.01 \times I_{p\times p}$	$\Sigma_{C_j} = diag([0.01 \;\; 0.0098]$	
$\boldsymbol{D_{1,j}}$	$\boldsymbol{\mu_{D_{1,j}}} = [0.15 \;\; 0.15$	$[-0.0145 \;\; 0.3141]$ (0.2361)	0.01853
	$0.15 \;\; 0.15]$	$[-0.0134 \;\; 0.3139]$ (0.2584)	0.0185
		$[0.0150 \;\; 0.3157]$ (-0.0227)	0.01863
		$[-0.145 \;\; 0.3150]$ (0.2216)	0.01865
	$\boldsymbol{\Sigma_{D_{1,j}}} = 0.01 \times I_{p\times p}$	$\Sigma_{D_j} = diag([0.009 \;\; 0.01192$	
		$0.01067 \;\; 0.01098])$	
Para.	Initial mean (μ)	[EM $\pm$95% conf. int.] (true value)	
$\boldsymbol{Q_{1,j}}$	$\alpha = 2 \;\; \beta = 1$	$[0.9365 \;\; 1.05]$ (1)	0.04197
		$[-0.098 \;\; 1.057]$ (1)	0.04268
	$\boldsymbol{\mu_{Q_{1,j}}} = I_{k\times k}$		
$\boldsymbol{R_{1,j}}$	$\alpha = 2 \;\; \beta = 1$	$[0.9839 \;\; 1.0177]$ (1)	0.04398
		$[0.9837 \;\; 1.0178]$ (1)	0.04255
		$[0.9834 \;\; 1.0182]$ (1)	0.04431
		$[0.9871 \;\; 1.0188]$ (1)	0.04276
	$\boldsymbol{\mu_{R_{1,j}}} = I_{p\times p}$		

Table 3.3: Initialization of hyperparameters and estimation of mean and covariance for the 1^{st} rows of parameter matrices **A**, **B**, **C**, **D**, **Q** and **R**.

get smoother towards the end of the chains. The trace plots for each parameter and the corresponding different runs are shown in Figure 3.8.

Once mixing of the Markov chain is confirmed by visual inspection, it is

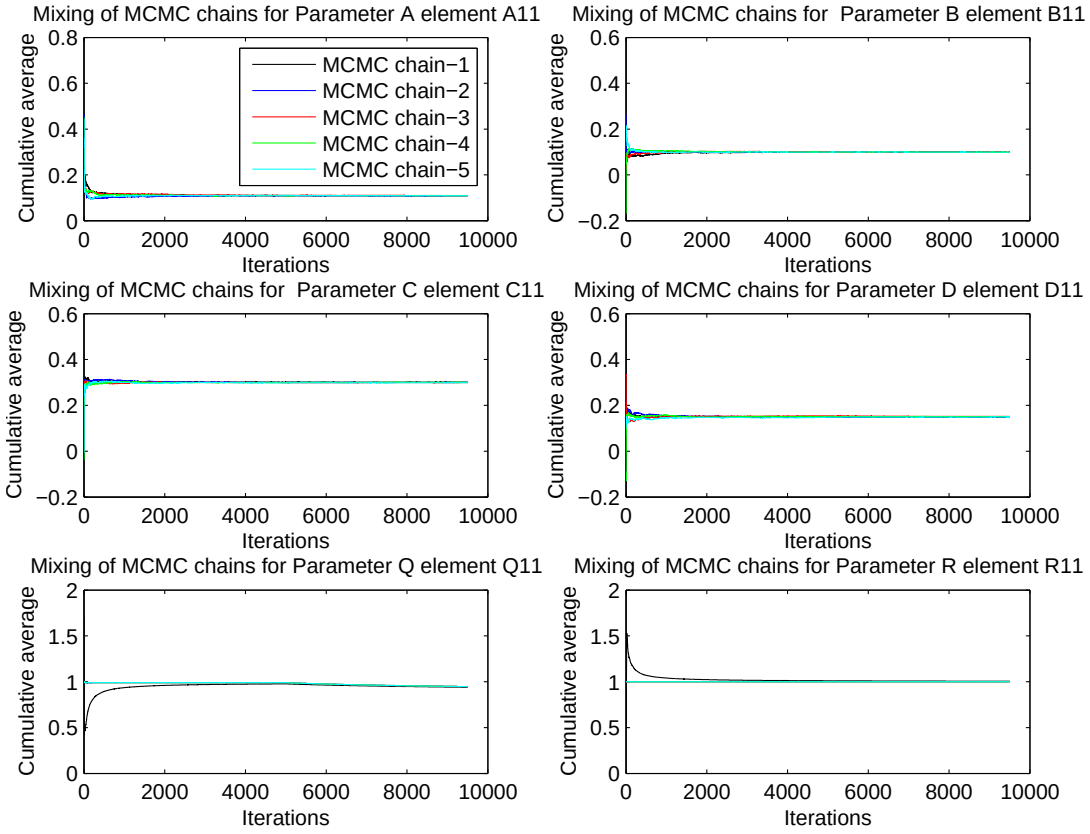


Figure 3.8: The visualisation of trace plots of different MCMC chains for the first element of SSM parameters **A**, **B**, **C**, **D**, **Q** and **R** i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11} and r_{11}

possible to calculate the convergence. For this reason we calculate the PSRF values once again (as described in Section 2.3.1) as shown in Table 3.4. We observe that most of the values are close to 1 or lie within an interval of $1.1 - 1.2$ demonstrating that the convergence criterion is satisfied here. Convergence seems to be improved in comparison with the parameter estimation given in the previous Subsection 3.3.2. Therefore it is ideal to move on to the next step which is to estimate the parameters.

Figure 3.9 shows the marginal posterior distributions from the M-H within Gibbs algorithm. Different colors of the density function represent five independent MCMC chains. The initial thick red curve is used to show the parameters of generating model that overlaps with the estimated distribution. The estimated mean

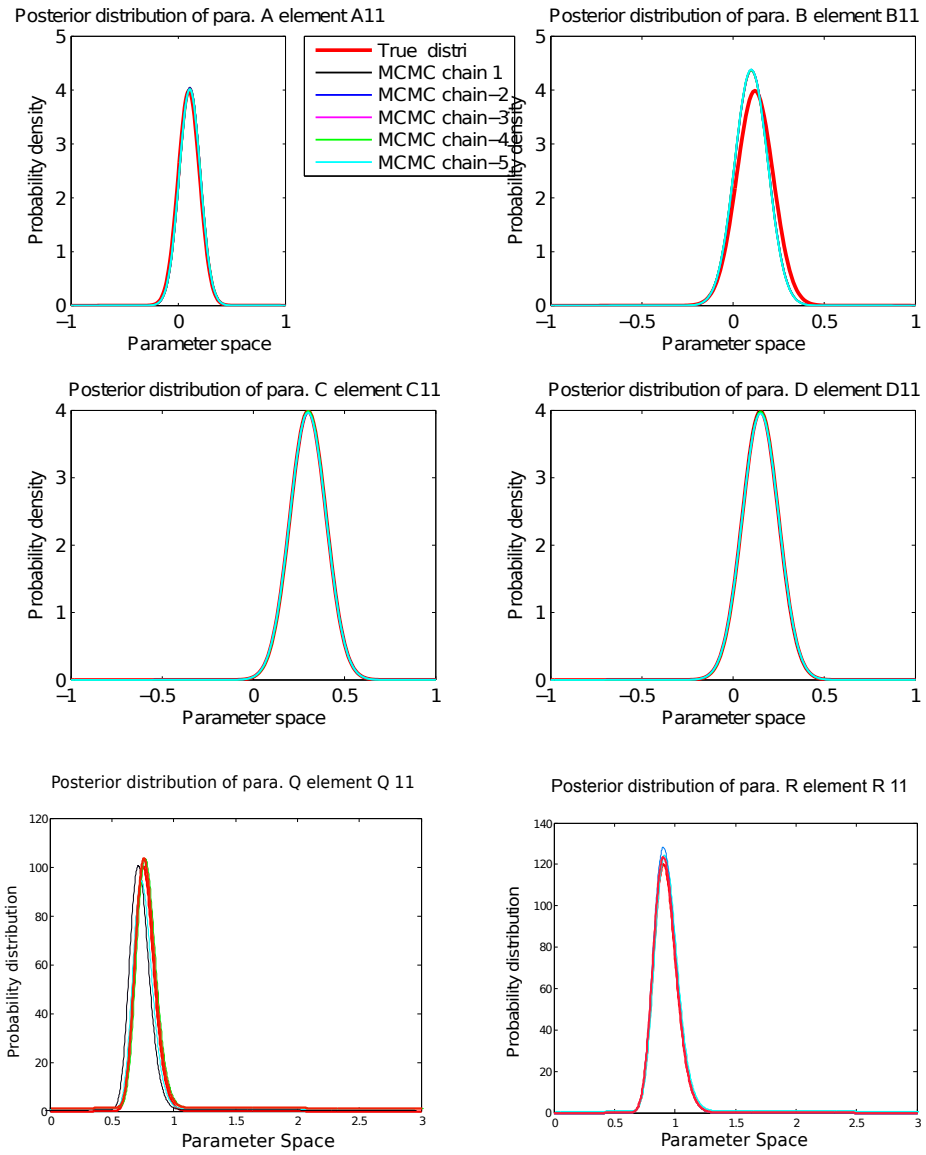


Figure 3.9: The marginal posterior distributions for different MCMC chains for the first element of SSM parameter matrices **A**, **B**, **C**, **D**, **Q** and **R** i.e. a_{11}, b_{11}, c_{11}, d_{11}, q_{11} and r_{11} from M-H within Gibbs algorithm.

and covariance from the Gibbs output are given in the second last columns of Table 3.3, which shows the mean of the parameters lie within 95% confidence interval. The low values of KL divergence calculated between the parameter distributions of the generating model and the marginal posterior distributions are reported in the last column of Table 3.3. KL divergence values indicate that the distribution of

PSRF	$A_{1,:}$	$B_{1,:}$	$C_{1,:}$	$D_{1,:}$	$diag(Q)_{2x1}$	$diag(R)_{4x1}$
	0.9957	1.0022	0.9973	0.9983	1.0533	0.99251
	1.0046	0.9638	1.0040	0.9964	1.0547	0.99328
		0.9944		0.9986		0.99560
		0.9912		0.9980		0.99477

Table 3.4: The PSRF for 1^{st} row of parameter matrix of SSM.

the generating model and the posterior distribution converges towards an identical distribution.

3.4 Simulating state and observation sequences using inferred parameters.

In this section we illustrate an approach of simulating observations by using the inferred parameters and hidden state sequence of an SSM. Section 3.3.1 describes how simulated data was generated using given set of parameter values, which are shown as in Hinton diagram here on the LHS of Figure 3.10. On the RHS of Figure 3.10 shows the Hinton diagrams of estimated parameters.

Following an algorithm 5 we randomly defined $\boldsymbol{x}_0$, $\boldsymbol{y}_0$ and estimated set of SSM parameters. We estimate $\boldsymbol{x}_{1:T}$ and $\boldsymbol{y}_{1:T}$ using algorithm 5. The algorithm is straightforward; firstly it generates samples $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ as shown in step (1) and (2) and then iterates it over given number of time points T.

Algorithm 5: Algorithm to regenerate observation and state sequences.

Input: Given randomly defined $\mathbf{x}_0$ and $\mathbf{y}_0$. Estimated parameter matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$, $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$, time point $T = 100$.

Output: Generate $\mathbf{x}$ and $\mathbf{y}$

1 Sample $\boldsymbol{x}_1 \sim N(\hat{\mathbf{A}}\boldsymbol{x}_0 + \hat{\mathbf{B}}\boldsymbol{y}_0, \hat{\mathbf{Q}})$

2 Sample $\boldsymbol{y}_1 \sim N(\hat{\mathbf{C}}\boldsymbol{x}_1 + \hat{\mathbf{D}}\boldsymbol{y}_0, \hat{\mathbf{Q}})$

3 **for** $t \leftarrow 2$ ***to*** T **do**

4 $\quad \boldsymbol{x}_t \sim N(\hat{\mathbf{A}}\boldsymbol{x}_{t-1} + \hat{\mathbf{B}}\boldsymbol{y}_{t-1}, \hat{\mathbf{Q}})$

5 $\quad \boldsymbol{y}_t \sim N(\hat{\mathbf{C}}\boldsymbol{x}_t + \hat{\mathbf{D}}\boldsymbol{y}_{t-1}, \hat{\mathbf{R}})$

6 **return** $\boldsymbol{x}_{1:T}$ $\boldsymbol{y}_{1:T}$

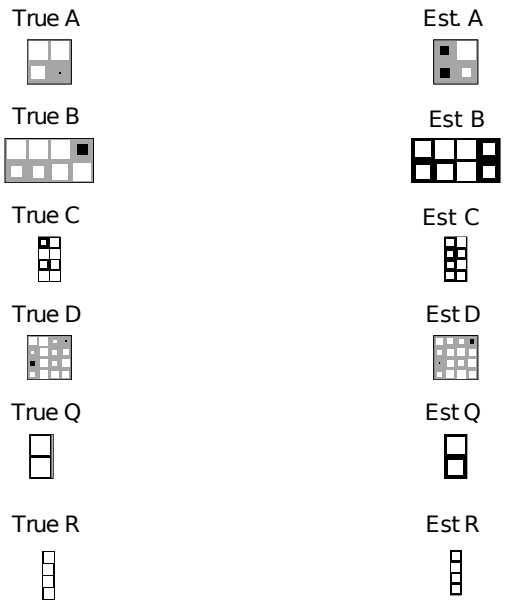


Figure 3.10: Hinton plots respresenting true parameters of the generating model on the LHS and estimated posterior means of parameters from MCMC output on RHS, for matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{Q}$ and $\mathbf{R}$

Using above algorithm we regenerate hidden state sequences $\mathbf{x_{k\times T}}$ and observation sequences $\mathbf{y_{p\times T}}$ as shown in Figures 3.11a, 3.11b, 3.12a, 3.12b. In each of figures from 3.11a-3.12b, the simulated sequence is ploted together with the true sequences (follow legend). The top subplot shows observation sequences and the bottom subplot shows the hidden state sequences. In order to avoid confusion only first dimension of the hidden state sequence is included here.

3.5 Summary

In this chapter we have demonstrated the validation of our algorithm by following a test algorithm. To check performance of our algorithm we have performed a numerical experiment using simulated data. This numerical experiment aims to recover the parameters of the generating model. It was observed that there was a lack of proper convergence due to improper mixing of the Markov chains which required

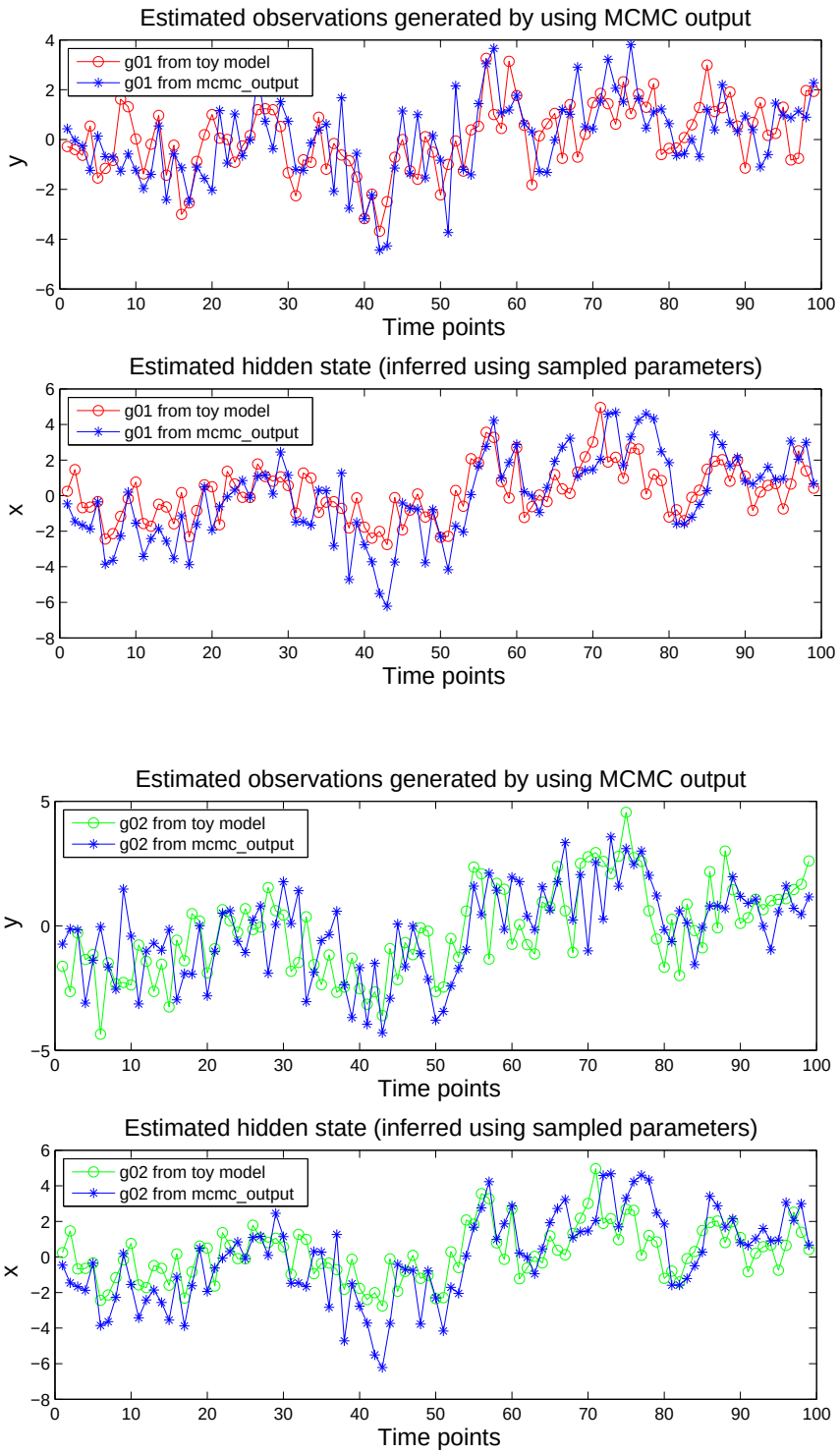


Figure 3.11: The reconstructed observation sequence (on top of the plot) and hidden state sequence (on bottom of the plot) using estimated parameter values those are shown on the RHS of figure 3.10.

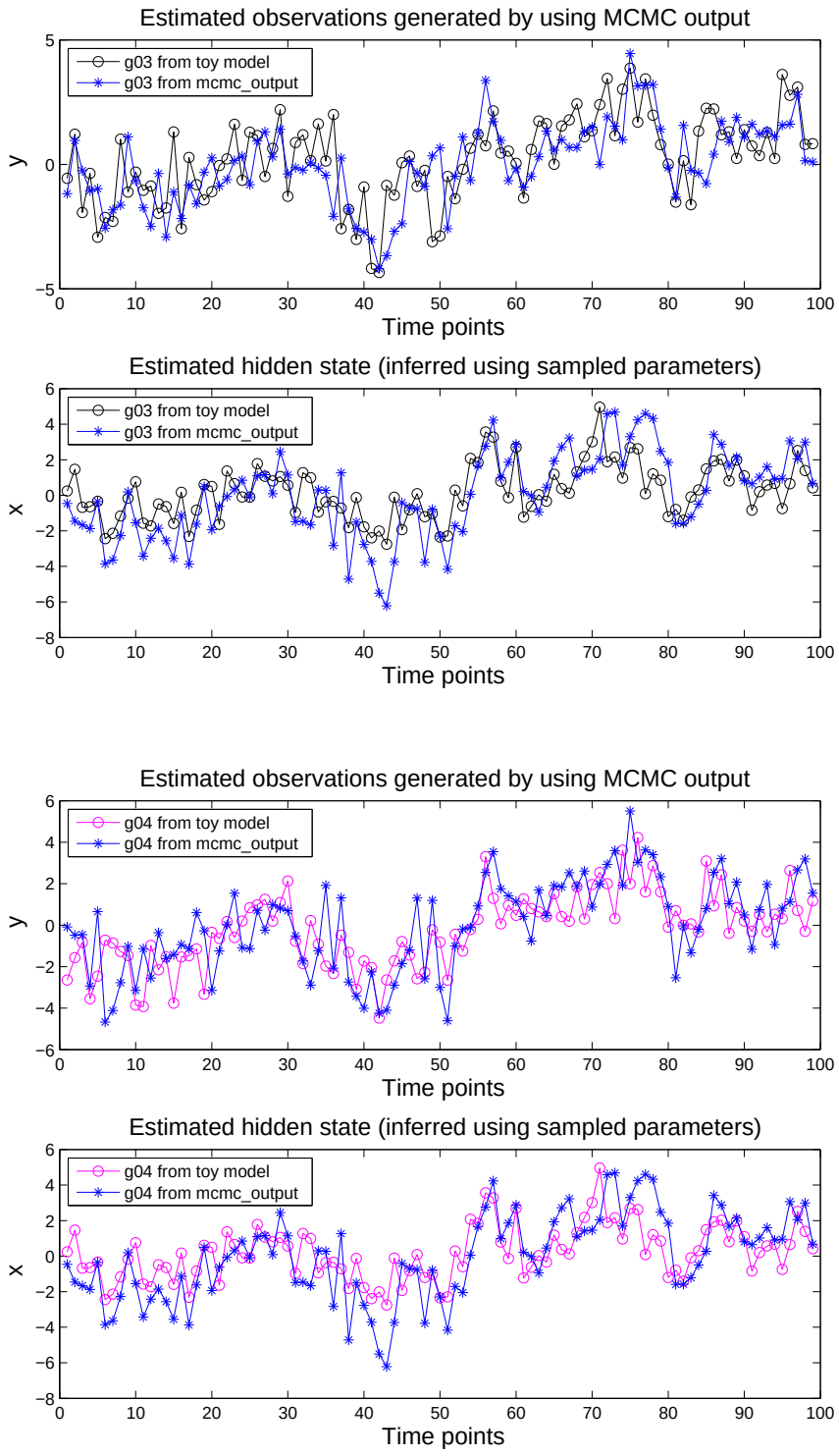


Figure 3.12: The reconstructed observation sequence (on top of the plot) and hidden state sequence (on bottom of the plot) using estimated parameter values those are shown on the RHS of figure 3.10.

some attention. This issue was addressed by introducing a M-H within Gibbs step to learn the covariance matrix of parameters **A**, **B**, **C**, **D** and hyperparameters of parameters **Q** and **R**. By inferring the hyperparameters we have observed in Section 3.3.3 that most of the parameters of the SSMs show proper mixing and demonstrate convergence. In this way we observe that the true parameters lie within the 95% confidence interval of the estimated parameters. In the section 3.4 we recapitulate gene expression observation sequence generated from an SSM.

This simulation was simple as the data were actually generated from a linear dynamical model. However additional care is needed while handling actual gene expression data. In reality the observations of gene expression are more noisy, and most likely require a hidden state space of larger dimensionality. In the following chapters we will show the application of our algorithm using a biologically plausible *in silico* dataset and later on application to experimental microarray datasets.

Chapter 4

Network inference to reverse engineer an *in silico* network

In this Chapter we test the performance of our algorithm using a more biologically plausible dataset. We aim to reverse engineer a genetic regulatory network by using simulated data from the *in silico* synthetic network proposed by Zak et al. [2003].

In general it is good practice to confirm the performance of proposed algorithms or techniques by using a variety of realistic datasets. Details of the network and how the data is produced is described in Section 4.1. Section 4.2 demonstrates the implementation and analysis of the Gibbs sampler. In Section 4.3 we explore the issue of model selection and discuss the selection of a suitable dimension for the hidden state space. In Section 4.4 (a) we describe a statistical hypothesis test to deduce the connection of the regulatory network from the inferred gene-gene interaction matrix. In Section 4.4 (b) we compare the network inferred from the Gibbs sampler with the variational Bayesian approach [Beal et al., 2005] and in Section 4.4 (c) we perform a receiver operating characteristic (ROC curve) analysis to assess the accuracy of the network reconstructed.

4.1 Background

Zak et al. [2003] present an *in silico* model based on mammalian cells, which constantly remodel their transcriptional activity profile in response to external input. The *in silico* genetic regulatory network shown in Figure 4.1 is constructed by arranging known modules of transcriptional regulation into regulatory motifs, using knowledge based on the biological literature.

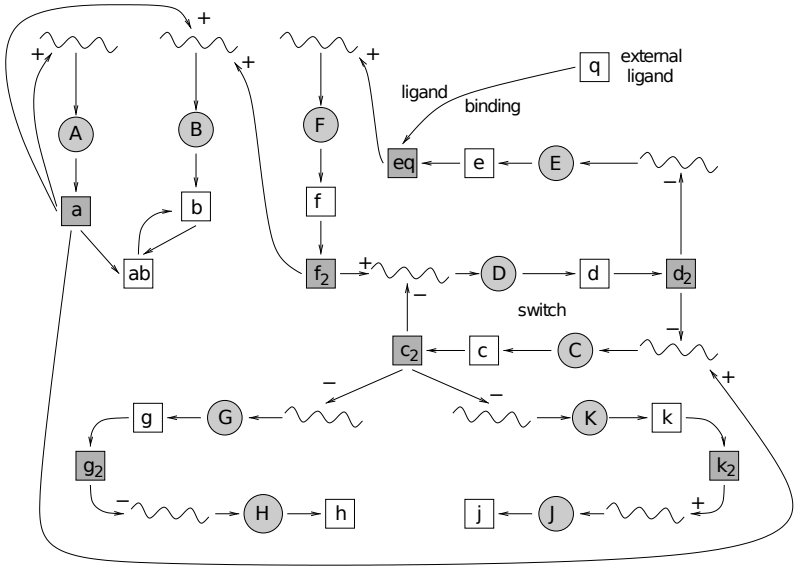


Figure 4.1: The *In silico* genetic regulatory network, adapted from Husmeier et al. [2005]. This is a network designated by letters to represent genes. The curly lines are the promoters, the circles show mRNAs and the squares represent proteins. Shaded squares are active transcriptional factors. A + sign shows a transcription factor acting as an activator and a − sign acting as an inhibitor. q is an external ligand and is used to introduce a switch in a network.

The overall structure of the network is chosen in a way that, in the absence of ligand input, there are high levels of mRNA for the transcriptional factors a and c, protein b, receptor e and the downstream protein h as shown in Figure 4.2. When the ligand q is introduced, the cell shifts into a state where the mRNAs for transcription factors a, c and receptor e are present at low levels and the mRNAs for transcription factors d and f and the downstream protein j are present at high

levels. When the ligand is removed, the cell returns to its initial state. The model parameters were selected to yield time scales representative of mammalian gene expression [Zak et al., 2003].

In silico simulations were carried out using the MATLAB code provided by Zak et al. [2003], which numerically integrates the deterministic ordinary differential equations (ODEs), describing the model. The model then simulates mRNA profiles as a constructed time series sampled over a length of $40h$ with 5 replicates. Gaussian noise was then added over the 10×40 gene expression measurements. By using this dataset we evaluate the performance of our algorithm.

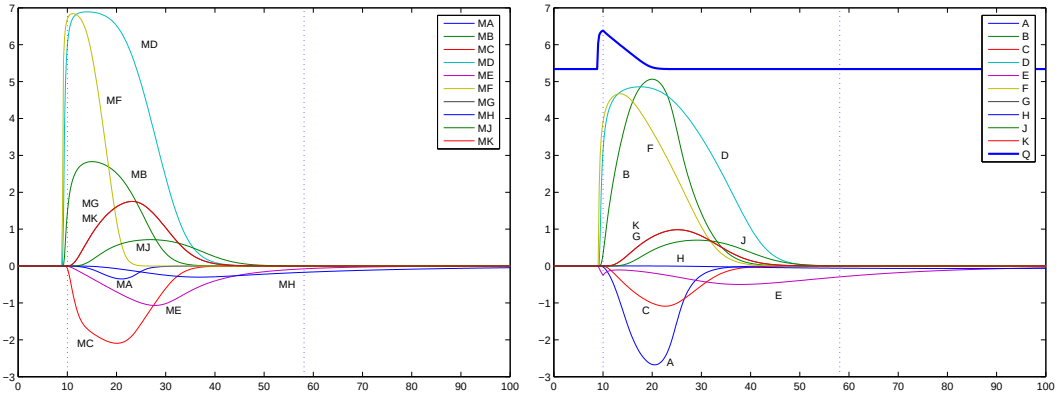


Figure 4.2: Represents the results of the deterministic ODE simulation profiles for 10 genes. The figure on the LHS shows the mRNA expression $MA, MB, \ldots, MK$ and the figure on the RHS shows the associated hidden protein levels $A, B, \ldots, K$. The dotted lines in both plots represent the time window of simulated data. In the RHS plot it can be observed that the time window begins at the peak of the injected ligand in the bold Q curve. This figure is adapted from Husmeier et al. [2005].

4.2 Numerical experiment

This section describes the numerical experiment. We began by simulating the Zak dataset of 48 time points with 5 replicates. The MCMC algorithm was then set to iterate from 4 different starting points for a sufficiently long time. In this experiment we chose the initial starting points randomly. As described in the previous chapter we investigated the convergence by visual inspection and measurement analysis of

the Markov chain for each of the model parameters **A**, **B**, **C**, **D**, **Q** and **R**. After convergence we calculated the marginal likelihood of the model using Chib's method as described in Chapter 2 and compared it to that calculated by a variational based approach.

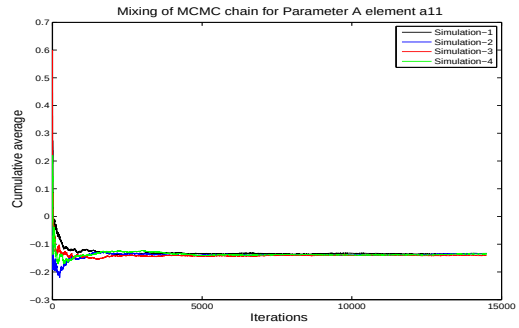
We ran the MCMC sampler for $150,000$ iterations, thinning by saving every $10th$ sample, in order to minimise correlation between consecutive samples drawn. We ran chains from four different random starting points. We discarded initial 5000 samples for parameters **A**, **B**, **C**, **D**, **Q** and **R** and carried out convergence diagnosis as in the following Section 4.2.1.

4.2.1 Diagnosis of convergence

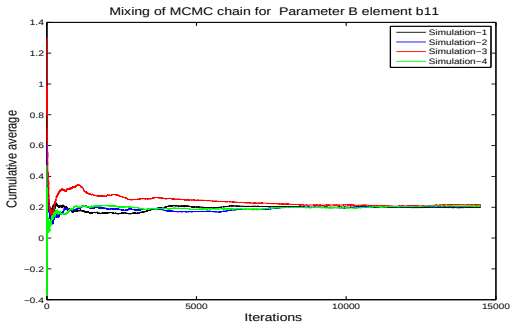
As an initial check on the drawn samples we compare the trace plots from four MCMC chains. Trace plots are shown in Figure 4.3 for first element of parameter matrices **A**, **B**, **C**, **D**, and **R**. The trace plots of **Q** and **R** shared a similar parameter space therefore we choose one of them to represent noise parameter. All other parameters are shown in the supplementary material. We observe that all different chains effectively converge to identical values and indicate that the chains explore a similar parameter space with proper mixing. We further confirm mixing and convergence by performing some formal tests.

Considering formal tests to confirm our beliefs, we have implemented some of the tests described in the Bayesian Output Analysis package (BOA), [Smith, 2007] using MATLAB. We have applied the Brooks and Gelman [1998] test to the four chains to check whether they converged to the same target distributions. As previously mentioned, the *PSRF* test compares the variance of each parameter within each chain with the variance between chains, giving a similarity measure. Figure 4.4 shows that the PSRF estimates are ≤ 1.2. Therefore the chains can be considered to have reached stationary distribution, as seen in Figure 4.4.

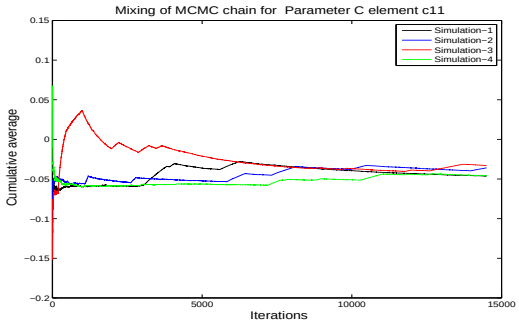
Another useful diagnostic are the kernel density plots (a.k.a. smoothed den-



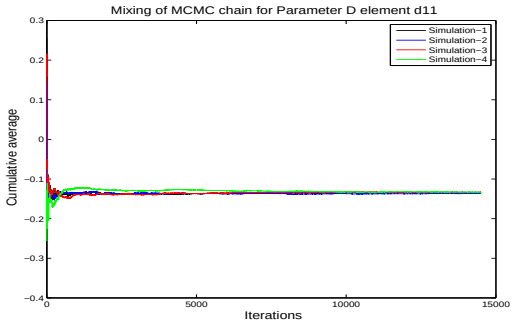
(a) Parameter A element $a(11)$



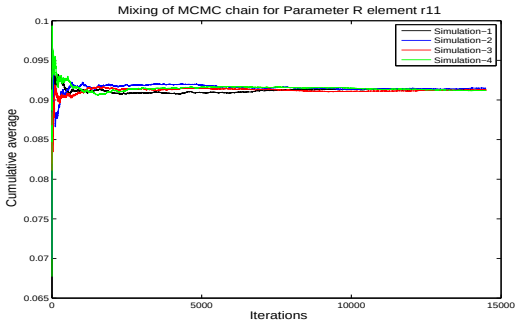
(b) Parameter B element $b(11)$



(c) Parameter C element $c(11)$



(d) Parameter D element $d(11)$



(e) Parameter R element $r(11)$

Figure 4.3: Trace plots of the first element of parameter matrices **A**, **B**, **C**, **D** and **R**.

sity plots). Sometimes non-convergence is reflected in multimodal distributions. This is especially true if the kernel density plot is not just multimodal but also ex-

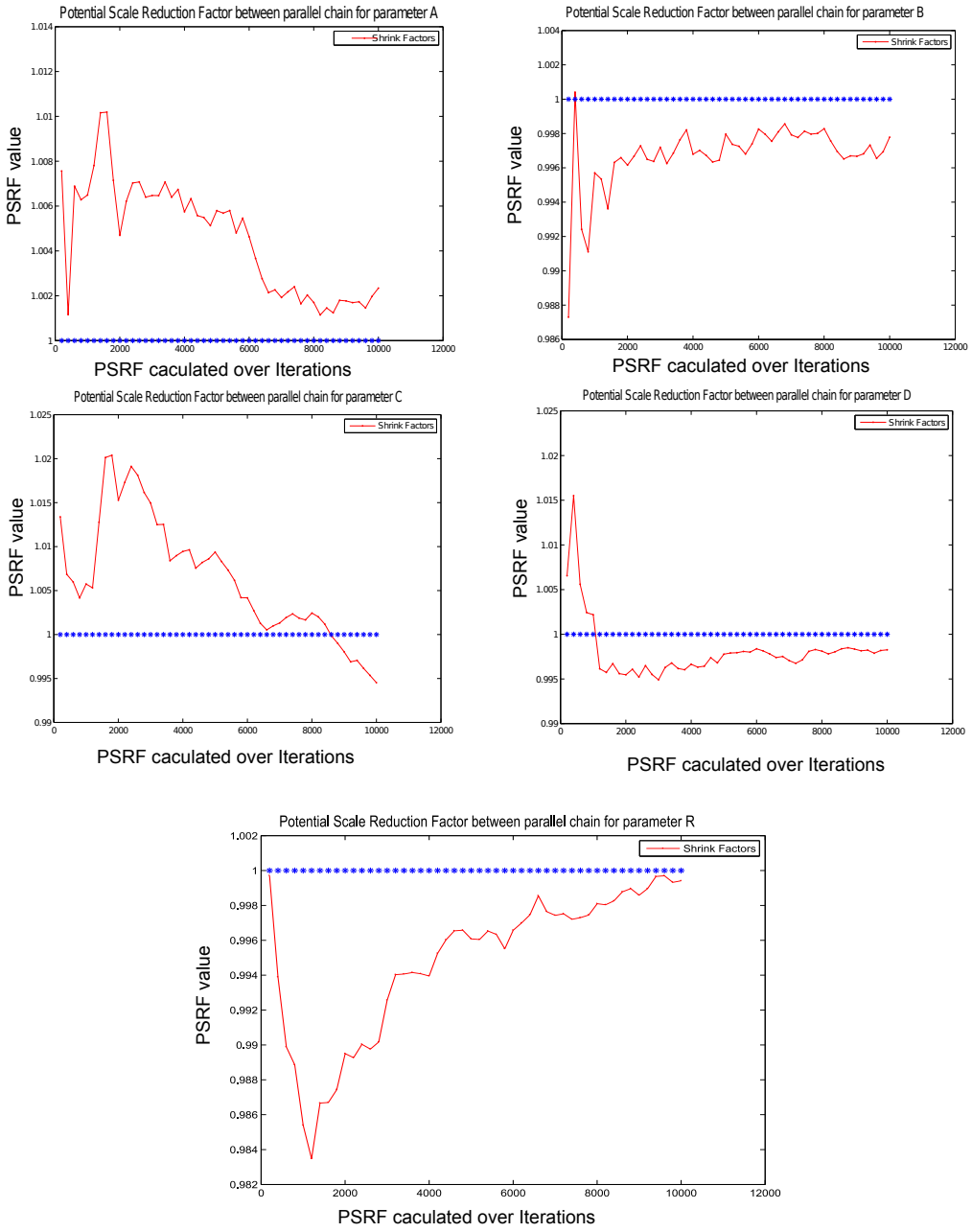


Figure 4.4: Potential scale reduction factor for the model parameters. The chosen parameters represent the first element of parameter matrices **A**, **B**, **C**, **D** and **R**. In this figure the PSRF value was calculated using binned intervals over entire iterations. The entire red curve shows the calculated PSRF values which are bounded between 0 and 1.2.

hibits nonlinear features. A kernel density plots in Figure 4.5 show that the posterior plots from the inferred parameters of different chains look similar.

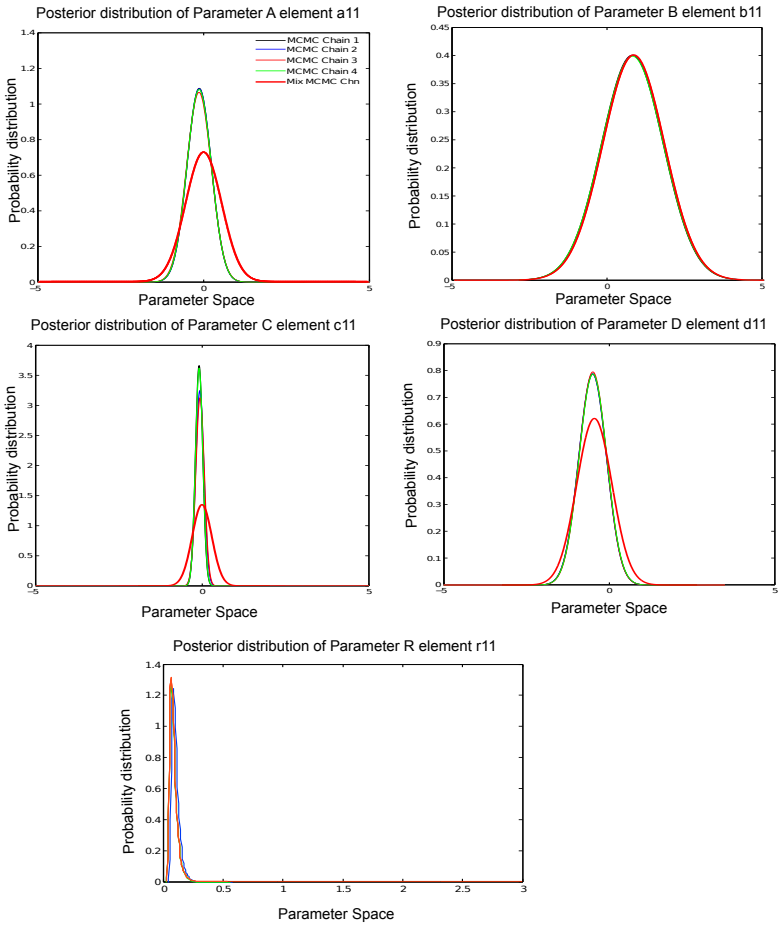


Figure 4.5: The kernel density plots of the first elements of marginal posterior parameter matrices **A**, **B**, **C**, **D** and **R**. The smoothed densities from different MCMC chains overlap, which indicates that the MCMC chains have converged to the same stationary distribution.

4.3 Model selection

4.3.1 Determination of state space dimensionality

The task of defining a suitable hidden state dimension is crucial and difficult. If we have defined too few hidden state dimensions then the model might not infer higher

order hidden dynamics. This might effect the estimation of gene-gene interactions. On the other hand, if we have defined too many hidden state dimensions, then more complex model may overfit the data and this will lead to ambiguous inferences.

In order to address these issues, Rangel et al. [2001] used cross validation experiments where part of the dataset was used to monitor the predictive likelihood. In this way one cannot make use of the entire dataset. In the variational Bayesian (VB) treatment approach of Beal et al. [2005] the entire dataset could be used to estimate both the parameters and identify an optimal hidden state dimensionality. In this work we make use of the sampler output to estimate parameters and in turn to find optimal hidden state dimensionality.

4.3.2 Calculation of model evidence

Model evidence was calculated from the Gibbs sampler output using Chibb's method (as described in Chapter 2 Section 2.4). Figure 4.6 shows the overall marginal likelihood calculations from the different MCMC runs with increasing hidden state dimension together with the variational Bayesian estimate from the VBSSM code [Beal et al., 2005]. In Figure 4.6 we observed decreasing marginal likelihood with increasing k (i.e., state space dimension) for this particular dataset. Therefore from this marginal likelihood study it is observed that for synthetic Zak data, hidden state space dimension of 1 is enough to model the data. A similar experiment was also performed by [Beal et al., 2005], where the authors also find the optimal dimension of the hidden state space for this dataset to be 1. In all cases the VBSSM algorithm calculates a lower bound on marginal likelihood as expected [Beal et al., 2005].

4.4 Reverse engineering the Zak network

After exploring the convergence of the model parameters and the estimation of an optimal hidden state dimensionality we are interested in the inference of the

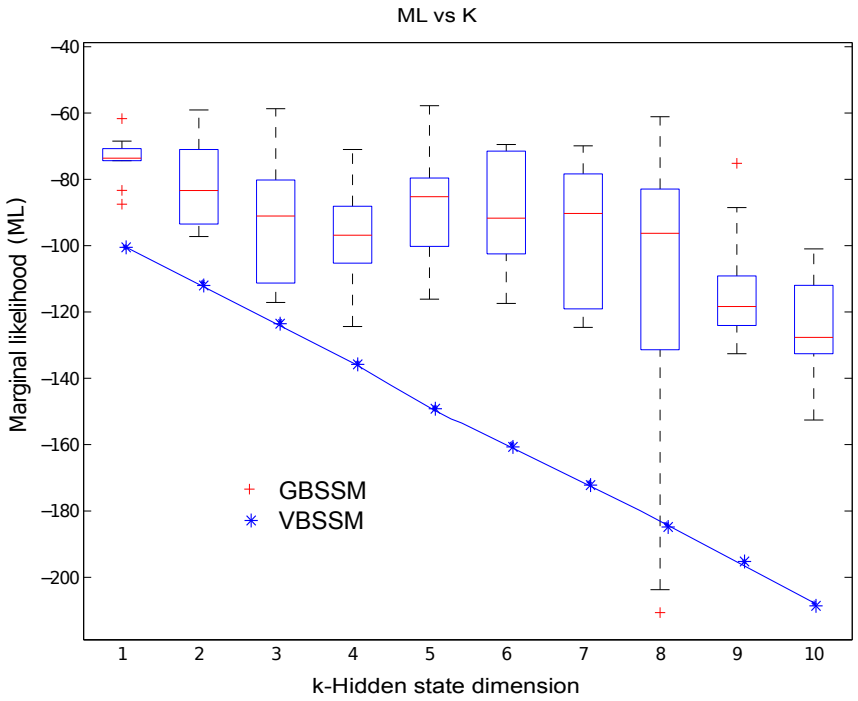


Figure 4.6: Comparing marginal likelihood from Gibbs sampling (GBSSM) and VBSSM approaches. VBSSM results form a lower bound to the estimates from the Gibbs sampler. The trend of the GBSSM results seems decreasing up to $k = 4$. Thereafter the increase in ML with $k > 4$ indicates the model may be over-fitting, while the trend of VBSSM shows a monotonic decrease which indicates that the model is not over-fitting the data.

connectivity matrix $[\mathbf{CB} + \mathbf{D}]$. We can obtain this matrix by substituting the state dynamic equation Equation 2.1 into the observation Equation 2.2, to yield:

$$\mathbf{y}_t = [\mathbf{CB} + \mathbf{D}]\mathbf{y}_{t-1} + [\mathbf{CB}]\mathbf{x}_{t-1} + (\mathbf{Cw}_t + \mathbf{v}_t) \tag{4.1}$$

where $[\mathbf{CB} + \mathbf{D}]$ from Equation 4.1 defines the *connectivity matrix*, also known as the gene-gene interaction matrix. $[\mathbf{CB}]$ represents the state to observation dynamics and $\mathbf{Cw}_t + \mathbf{v}_t$ is the combined noise of the model. The connectivity matrix can be estimated by using the estimates of parameters $\mathbf{C}, \mathbf{B}$ and $\mathbf{D}$ individually. This can be achieved either by the use of the Rao-Blackwellization theorem [Blackwell,

1947] or by carefully implementing the Gaussian sum and product rules. Once the connectivity matrix is estimated the hypothesis test described in the following Section 4.4.1 defines the interaction matrix. Finally we will compare our inferred network to those obtained from the variational approach.

4.4.1 Estimation and interpretation of connectivity matrix

We examine the gene-gene interactions as represented by the matrix $[\mathbf{CB}+\mathbf{D}]$. The MCMC algorithm provides the posterior distribution for each of the parameters. By using the marginal posterior distribution of the parameters we compute the connectivity matrix. As defined earlier the parameters $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and $\mathbf{x}$ are multivariate Gaussian distributed with estimated mean and covariances. The product of two Gaussians is an un-normalised Gaussian which means that we need to calculate the normalisation constant for calculating the product of $\mathbf{C}$ and $\mathbf{B}$. Following Appendix-A of Rasmussen and Williams [2006] we can denote the estimated posterior mean and covariance of parameter $\mathbf{C}$ from the sampler output as $\boldsymbol{\mu_C}$ and $\boldsymbol{\Sigma_C}$ respectively, and similarly the estimated posterior mean and covariance of parameter $\mathbf{B}$ to be $\boldsymbol{\mu_B}$ and $\boldsymbol{\Sigma_B}$. The mean and covariance of the product $\mathbf{CB}$ is denoted by $\boldsymbol{\mu_{CB}}$ and $\boldsymbol{\Sigma_{CB}}$, given by

$$N(\mathbf{C}|\boldsymbol{\mu_C}, \boldsymbol{\Sigma_C})N(\mathbf{B}|\boldsymbol{\mu_B}, \boldsymbol{\Sigma_B}) = Z^{-1}N(\mathbf{CB}|\boldsymbol{\mu_{CB}}, \boldsymbol{\Sigma_{CB}}),$$

where

$$\boldsymbol{\mu_{CB}} = \boldsymbol{\Sigma_{CB}}(\boldsymbol{\Sigma_C}^{-1}\boldsymbol{\mu_C} + \boldsymbol{\Sigma_B}^{-1}\boldsymbol{\mu_B}),$$

$$\boldsymbol{\Sigma_{CB}} = (\boldsymbol{\Sigma_C}^{-1} + \boldsymbol{\Sigma_B}^{-1})^{-1},$$

and

$$Z^{-1} = (2\pi)^{-D/2}\det(\boldsymbol{\Sigma_C} + \boldsymbol{\Sigma_B})^{-1/2}\exp\{-1/2(\boldsymbol{\mu_C}-\boldsymbol{\mu_B})^T(\boldsymbol{\Sigma_C}-\boldsymbol{\Sigma_B})^{-1}(\boldsymbol{\mu_C}-\boldsymbol{\mu_B})\}.$$

It is straightforward to find the sum of two Gaussians and this way we calculate each element of the combined matrix $[\mathbf{CB} + \mathbf{D}]$,

$$N(\mathbf{CB}|\boldsymbol{\mu_{CB}}, \mathbf{\Sigma_{CB}}) + N(\mathbf{D}|\boldsymbol{\mu_{D}}, \mathbf{\Sigma_{D}}) = N(\mathbf{CB} + \mathbf{D}|\boldsymbol{\mu_{CB+D}}, \mathbf{\Sigma_{CB+D}}),$$

where

$$\boldsymbol{\mu_{CB+D}} = \mathbf{\Sigma_{CB}}(\mathbf{\Sigma_C}^{-1}\boldsymbol{\mu_C} + \mathbf{\Sigma_B}^{-1}\boldsymbol{\mu_B}) + \boldsymbol{\mu_D}$$

$$\mathbf{\Sigma_{CB+D}} = (\mathbf{\Sigma_C}^{-1} + \mathbf{\Sigma_B}^{-1})^{-1} + \mathbf{\Sigma_D}$$

Considering each element of the combined matrix $[\mathbf{CB} + \mathbf{D}]$, we can identify the gene-gene interaction: if the mean of marginal distribution of that element is significantly close to the value zero then it is counted as having no influence. We can specify a significance value by considering if the zero point is more than n standard deviations from the posterior mean for that particular value (or element of the interaction matrix)[Beal et al., 2005]. By using Z statistics we can put a threshold on the normally distributed variables of the connectivity matrix. Since these marginal distributions are Gaussian, the location of the posterior mean will lie above or below the value zero, which will correspond to the positive or negative regulation. Considering this to be a simple decision problem with two hypothesis:

$$H_0 : [\mathbf{CB} + \mathbf{D}]_{i,j} = 0,$$

which shows no connection, and:

$$H_1 : [\mathbf{CB} + \mathbf{D}]_{i,j} \neq 0,$$

which shows a connection between i^{th} and j^{th} gene.

After putting threshold on the connectivity matrix $[\mathbf{CB} + \mathbf{D}]$ we obtain a directed graph where the diagonal elements show self interaction.

4.4.2 Comparison to the Variational Bayesian method

Using the same simulated dataset with $t = 40$ time points and 5 replicates we have also used the variational Bayesian algorithm proposed by [Beal et al., 2005] to infer parameters and hidden states. The variational based method is an approximation technique that calculates a lower bound on the marginal likelihood of the parameter space. However implementation of any MCMC technique will explore the entire parameter space and will infer a distribution over parameters and hidden states. This will certainly make a difference in the calculated marginal likelihood. It can be observed in Figure 4.6 that the marginal likelihood from the Gibbs sampler lies above the marginal likelihood calculated from the variational approach.

Hinton diagram representation of reconstructed network

The output of the significant elements of the connectivity matrix from the VBSSM and the Gibbs sampler output is shown below using a Hinton diagram 4.7. The Hinton diagram is a qualitative display of the elements of the data matrix, where every element is represented by a square box whose size represents the magnitude and white/black represents the $+/-$ sign respectively. In Figure 4.7 we compare the true Zak network to the inferred network by calculating significant elements in the gene-gene interaction matrix $[\mathbf{CB} + \mathbf{D}]$ from both the variational approach and from the Gibbs sampler for the significance threshold of 95% i.e z-score of 1.96 standard deviations (std). Each network was inferred from the state space model with different state space dimensions $k = 1, \ldots, 10$ (more results are in the supplementary material). The Hinton representation is good for visual comparison, however due to the large number of false positive or negatives observed in these plots we use receiver operating characteristic (ROC) and area under the curve (AUC) analysis to provide a quantitative analysis of performance.

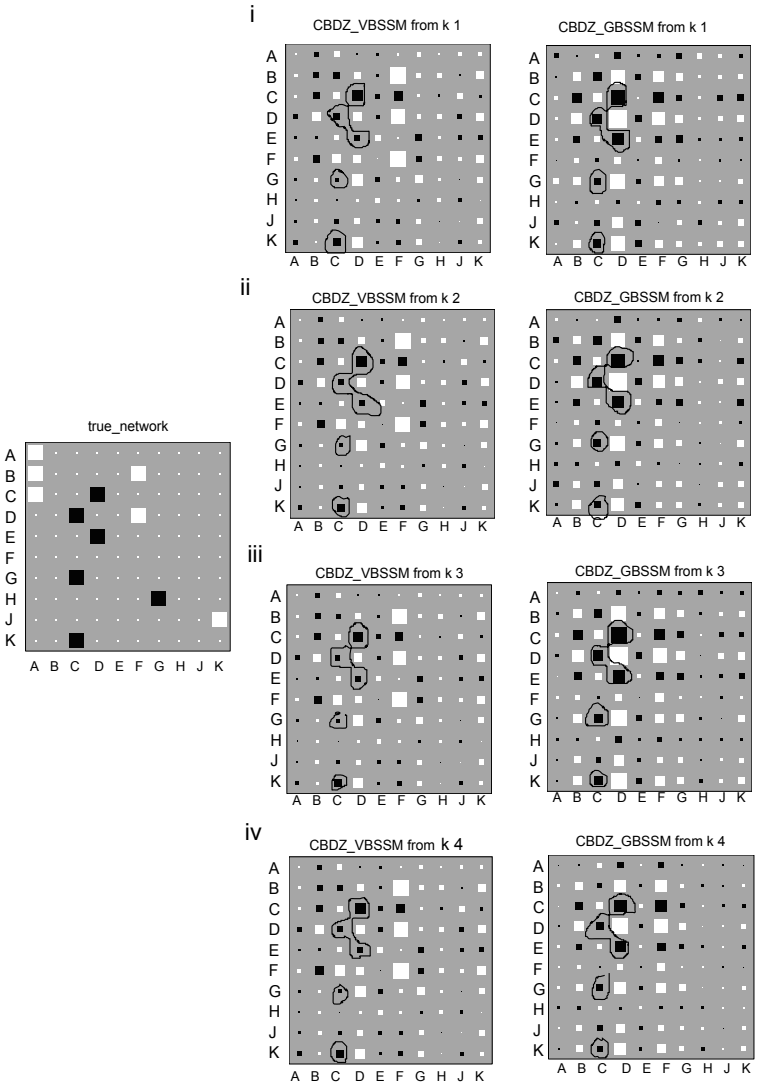


Figure 4.7: Hinton plots of the true network(on the left) against inferred networks from the variational method (VBSSM) and Gibbs sampler (GBSSM) for the significance threshold of 95% i.e z-score of 1.96 std. Panels i, ii, iii, iv represent the inference for hidden state space dimension $k = 1, 2, 3, 4$ respectively. The connectivity matrix designates genes $A, B, C, D, E, F, G, H, J, K$ on the x-axis and shows $+ve/-ve$ interactions between gene pairs as white and black squares. Islands are marked in the plots for easier comparison between identified elements. From the intensity of the dark blocks we can see the $-ve$ strength of the interactions from the GBSSM is much higher in comparison to VBSSM.

4.4.3 ROC and AUC analysis

This part of the analysis performs a ROC analysis (also known as a sensitivity versus specificity analysis) by using the known *in silico* gene regulatory network of Zak et al. [2003] as the "truth". ROC analysis can be done by calculating the sensitivity and (1-specificity) for different confidence levels considered when testing particular connections. The specificity is defined as the proportion of the recovered true network and the specificity is defined as the proportion of correctly identified non-connections,

$$\text{Sensitivity} = TP/(TP + FN),$$

$$\text{Specificity} = TN/(TN + FP).$$

Here TP is the true positive rate of actual connections that are considered to be connections. FP is the false positive rate of non-connections that were considered as connections. FN is the false negative rate of actual connections that were considered non-connected. TN is the true negative rate of non-connections which were considered to be non-connected. A perfect recovery of the network corresponds to a sensitivity and specificity of 1. In our consideration of true positives we also included the correctly interacting directions, which makes it a strict selection of TP compared to various previous studies.

In the following ROC plots shown in Figure 4.8, each point on the curve is computed based on the confidence levels ($z-$ score of higher than 1). The left figure shows the ROC analysis from the Gibbs sampler and VBSSM output for hidden state space dimension $k = 1$. In this figure different colors of curves represent different MCMC chains, along with the averaged ROC curve from the VBSSM, which is the thick red curve. Similarly on the right side the ROC analysis appears for hidden state space dimension $k = 4$. Other ROC curves for higher dimensions are provided in the supplementary material.

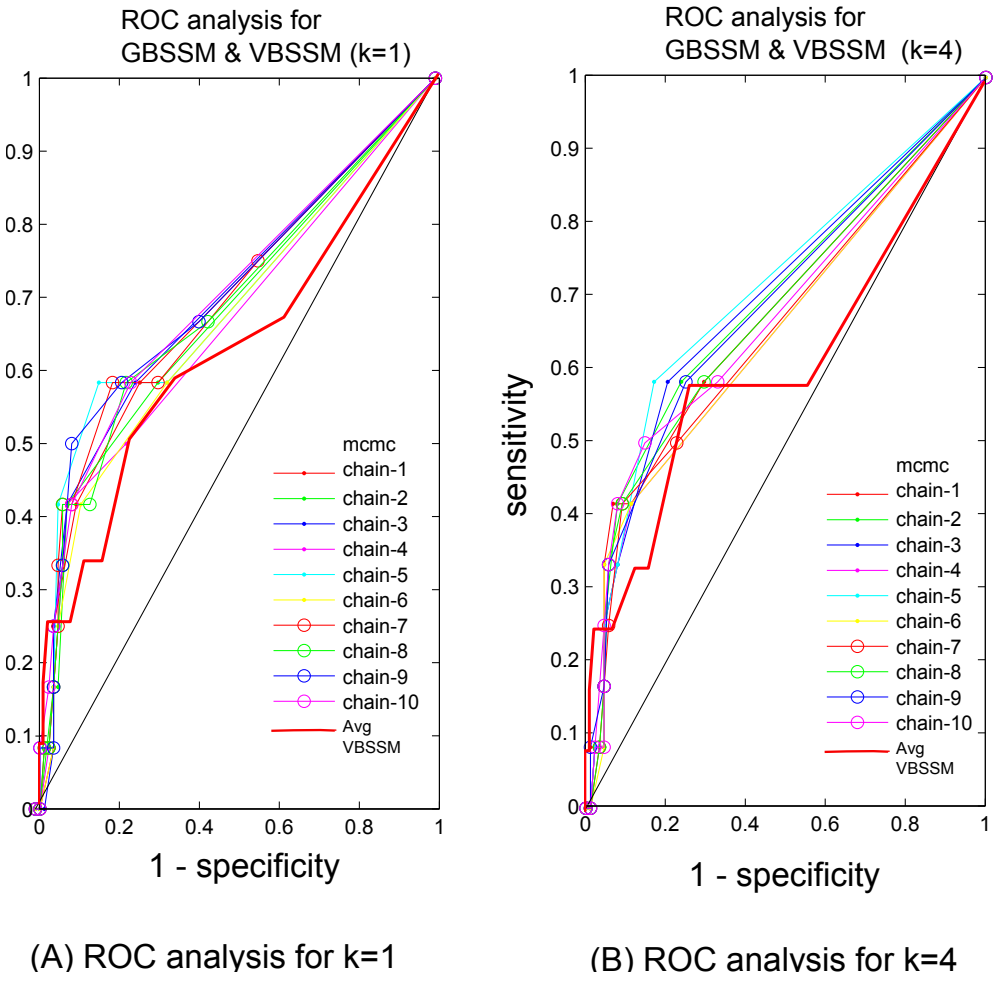


Figure 4.8: ROC curves calculated from VBSSM and GBSSM. Panel A shows the ROC analysis for the estimates from the GBSSM for hidden dimension $k = 1$. Different MCMC chains for the Gibbs sampler are represented by different curves as shown in the legend. The superimposed thick dark red curve represents the ROC curve calculated from VBSSM, which is the average over 5 different VBSSM simulations. Similarly Panel B shows the ROC analysis for a hidden state dimension of $k = 4$.

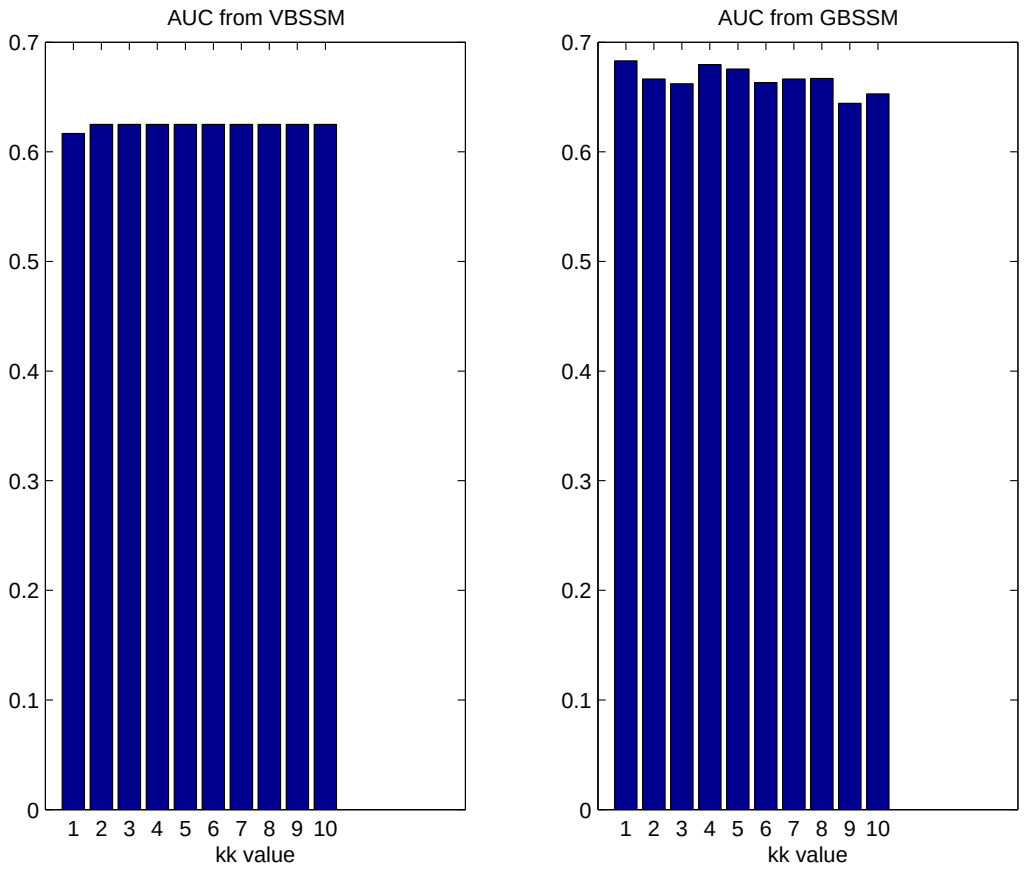


Figure 4.9: This plot represents the Area Under the Curve (AUC) for different values of the state space dimension k. The left plot is the AUC estimated from VBSSM and the right plot is from GBSSM. Here $k = 1$ represents the optimal state space dimension as found by model selection.

The Area under (ROC) curve or AUC is a useful statistics that is associated with ROC curve. As we known that the curve lies within the unit square, we have $0 \leq AUC \leq 1$. When $AUC = 1$ it indicates every TP is higher than every TN; when $AUC = 0$ then every TN is higher than every TP. The Area under the curve (AUC) calculated from VBSSM and GBSSM for hidden state space dimensions varying from $k = 1, 2, \ldots, 10$ is shown in Figure 4.9. We observe that for $k = 1$ the GBSSM achieves the highest value of the AUC i.e. 0.69, whereas the AUC from the VBSSM remains constant at a value of around 0.61. Both AUC is above 0.5 which shows that the performance of both algorithm is not performed randomly.

4.5 Regenerating *In silico* observations

Figure 4.10 shows the gene expression profiles for 10 genes from the *in silico* network as described in the next chapter. These expression profiles were simulated by solving a set of ODE's provided as a matlab tool by Zak et al. [2003]. After applying the Gibbs sampling algorithm, the parameters of an SSM i.e. $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{D}}$, $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ were inferred for a SSM with hidden state dimension $k = 1$.

Folllowing algorithm 5, the observation sequences were simulated using the inferred parameters and randomly defined hidden state $\boldsymbol{x}_0$ and observation $\boldsymbol{y}_0$ at time point $t = 0$. The simulated 10 gene expression profiles are shown in Figures 4.11a-4.13d. In Figure 4.11a and 4.11b the top subplot shows the simulated gene expression profiles and the inferred hidden state sequences $\mathbf{x}$ are shown in the bottom of the subplot. The inferred hidden state $\mathbf{x}$ remains same for all the simulated observation sequences, seen in the Figure 4.12a-4.13d.

4.6 Summary

In this chapter we have compared two techniques for parameter and hidden state inference for the SSM. The variational based approach is an approximate method

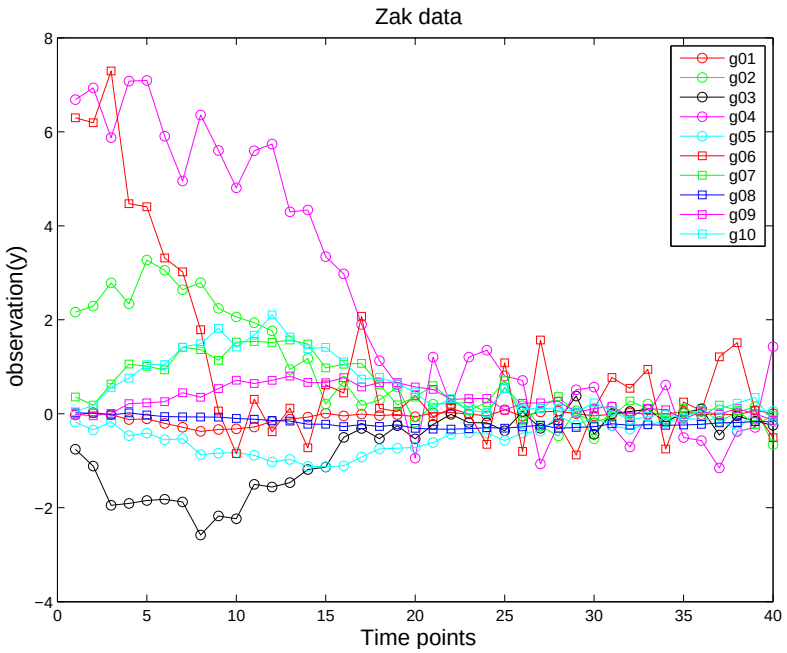


Figure 4.10: Simulated 10 gene expression profiles as described in Section 4.1.1.

which uses an EM algorithm. In the Gibbs sampler (MCMC) based approach, like the VBSSM, we have made use of the complete dataset. As the posterior distributions belongs to the conjugate exponential family it has allowed us to write down full conditional distribution of a Gibbs sampler for the inference of hidden state sequence and the parameter.

After implementation of both VBSSM and GBSSM algorithm we have observed that the marginal likelihood estimates from the Gibbs sampler includes a lower bound estimates of the VBSSM. The proposed GBSSM algorithm converges to a tread off point where increase in the number of hidden state space dimension might indicate identifiabilty issues. Nevertheless for an *in silico* dataset both algorithm shows the optimal state space dimension of $k = 1$ by model selection. Where GBSSM gains AUC of 0.69 of true positive interactions in comparison with AUC of 0.61 of true positives from VBSSM.

In the section 4.5 we can recapituate observation sequences generated from a

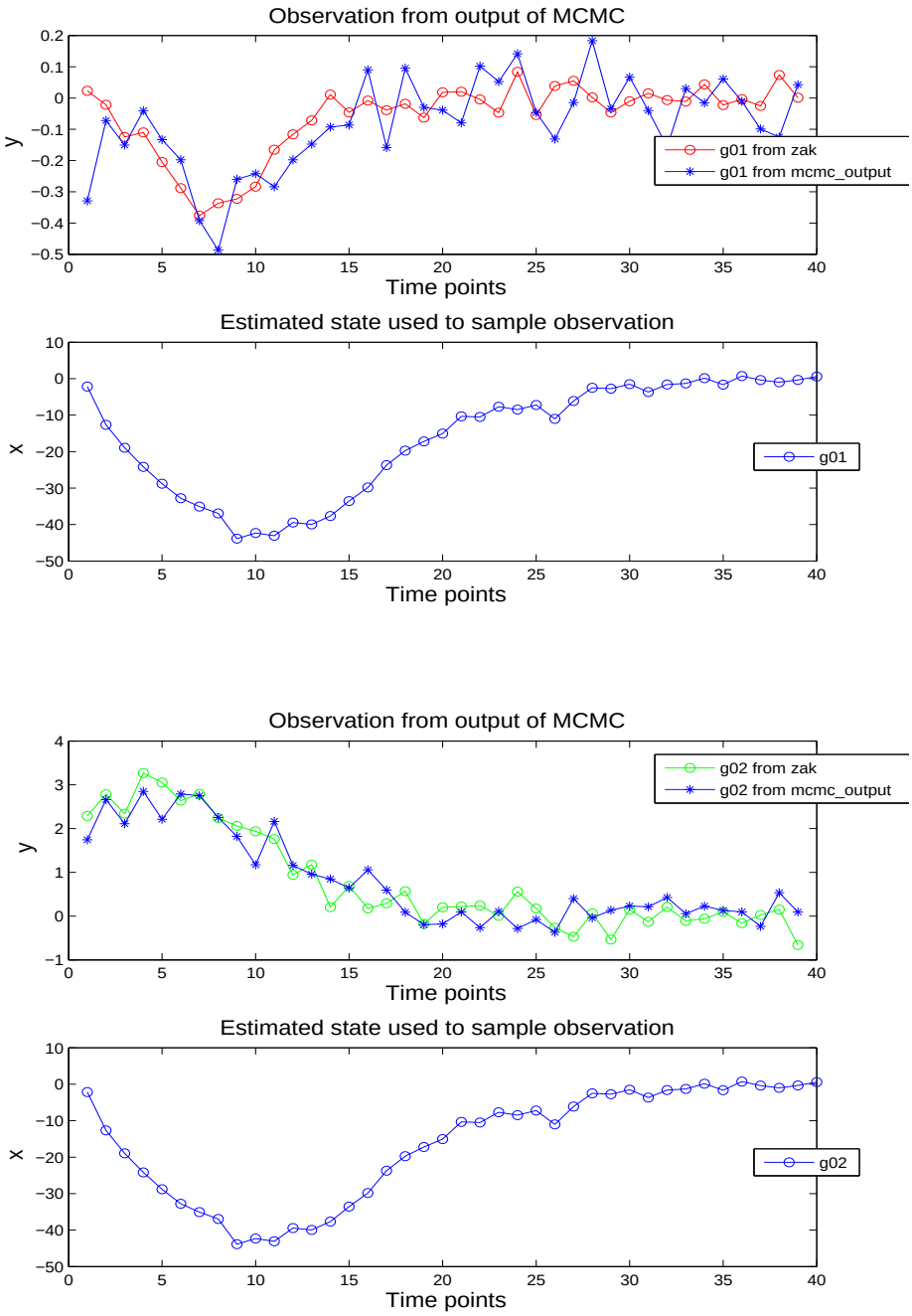


Figure 4.11: These plots shows simulated observation sequence on the top and inferred hidden state of dimension $k = 1$ on the bottom.

more realistic *in silico* ODE model of gene expression. Given estimated parameteres and inferred hidden state from Gibbs sampler output with the help of state space

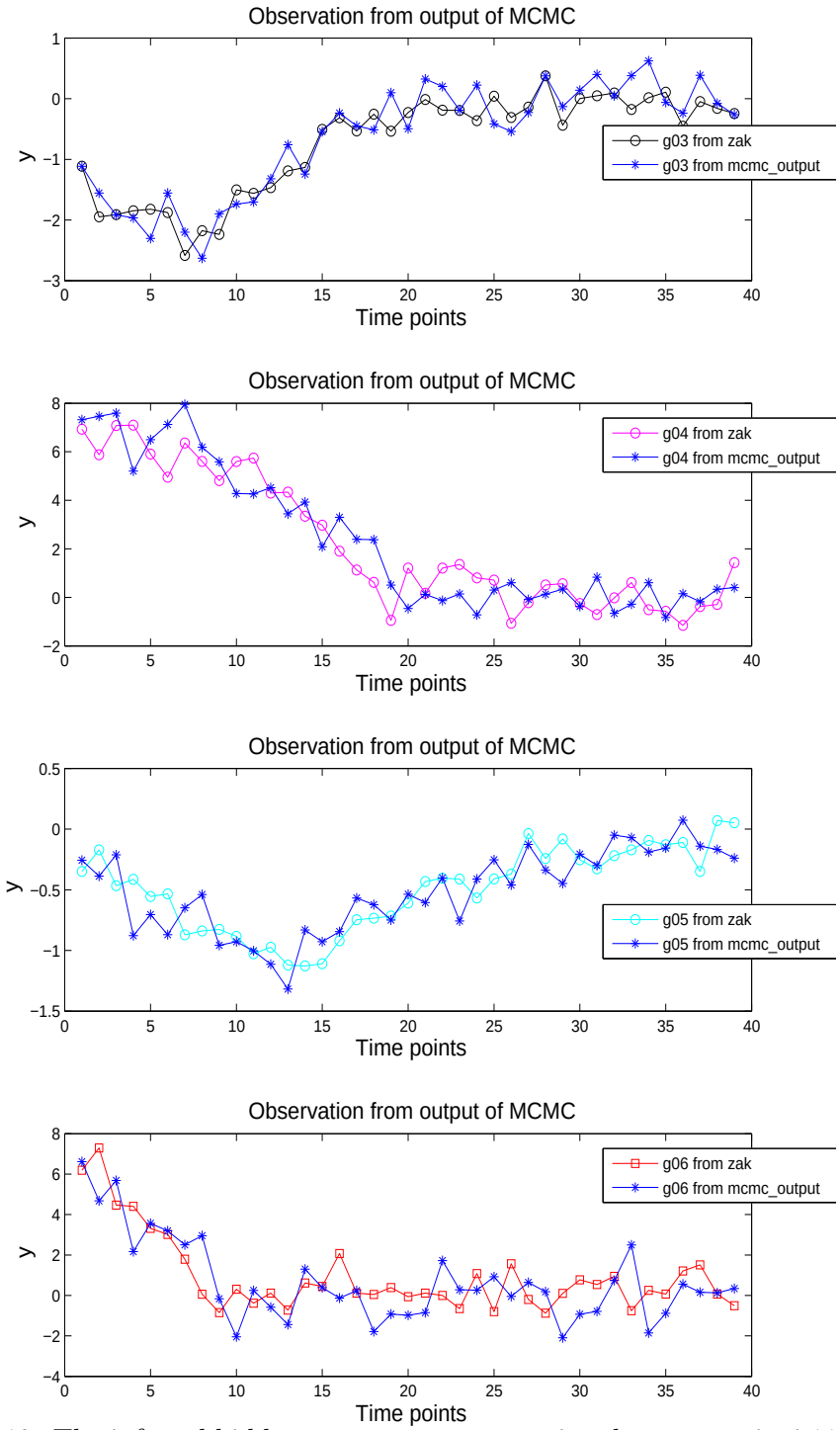


Figure 4.12: The inferred hidden state sequence remains the same as in 4.11a therefore only simulated observation sequence are shown here. Following the legend simulated and *in silico* observation sequence is shown in the same plot.

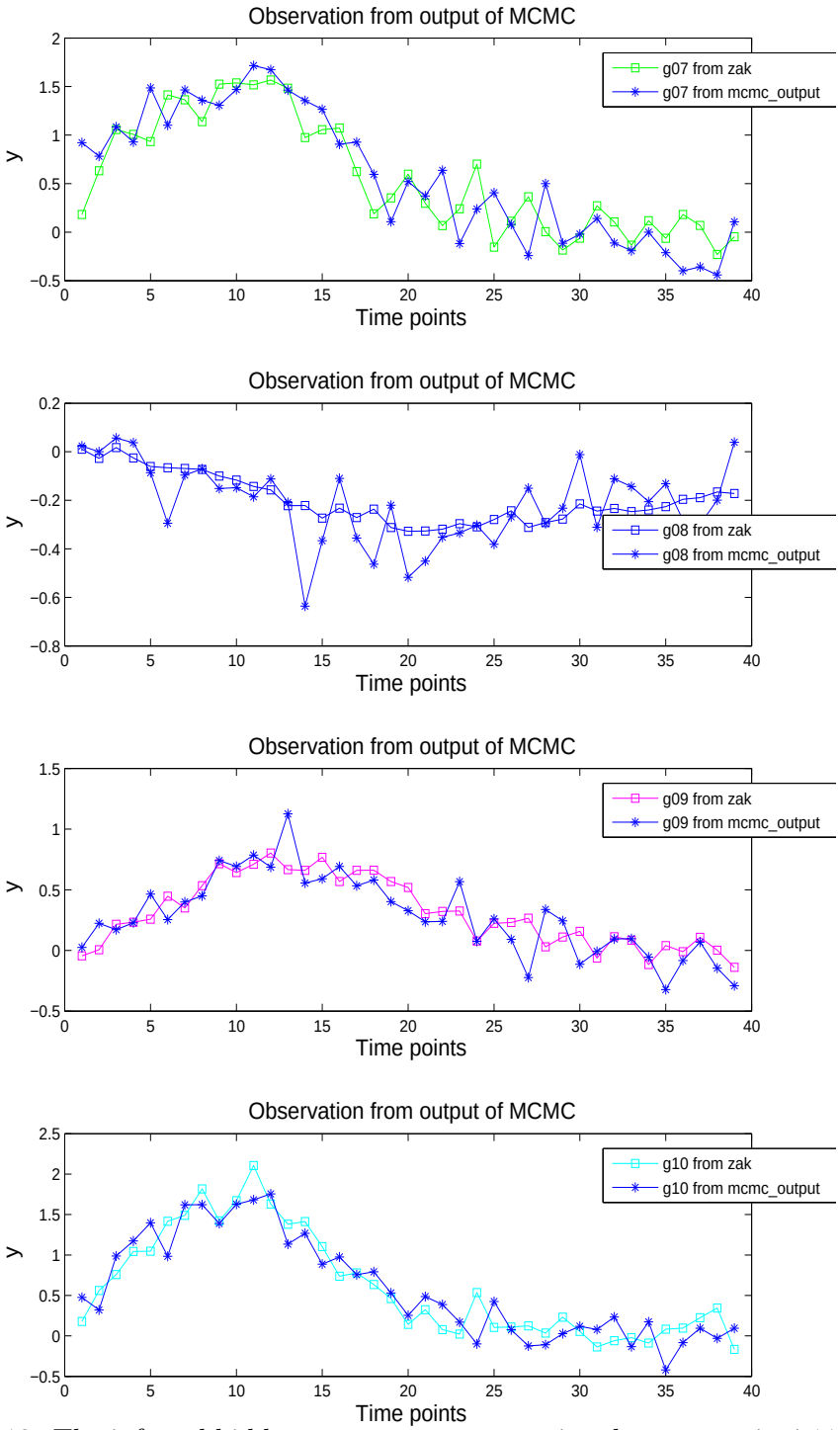


Figure 4.13: The inferred hidden state sequence remains the same as in 4.11a therefore only simulated observation sequence are shown here. Following the legend simulated and *in silico* observation sequence is shown in the same plot.

equations we have simulated observation sequences, indicating optimal state space dimension of $k = 1$ provides the best learning of parameters and hidden state space. This particular approach of validation is unique and as we can see it unfolds the behaviour of the estimated parameters and hidden state sequence quite well.

Chapter 5

Application to microarray data: The adaptation of *E. coli* cells to temperature shift (between $10°C - 37°C$)

Most living organisms encounter continuous environmental changes. Temperature change is one factor that has a profound effect on cell life through biochemical perturbations. However, variation in temperature also affects the structural and functional properties of cellular components. As a result it might also affect the cell's most essential processes such as replication, transcription, translation and membrane biogenesis. The chosen example in this case study covers the adaptation of *E. coli* cells to a temperature shift from $10°C$ to $37°C$. The objective here is to provide an essential understanding of bacterial physiology, regulatory networks and the underlying molecular mechanisms.

5.1 Biological background

The model organism chosen in this study is *Escherichia coli* K-12 of strain $MG1655$. *E. coli* is an excellent organism for systems biology. Its rapid growth rate under simple nutritional conditions makes experiment validation easier than other organisms. There is better established genetic and complete genome sequence information available for *E. coli* than for any other living organism. Many strains of *E. coli* have been sequenced and studied in detail. The article by Blattner et al. [2007] presents the complete $4,639,221$ base pair sequence of *E. coli* K-12, also including 4288 annotated protein-coding genes. However, 38% of them had no assigned function at the time of publication. *E. coli*, as shown in Figure 5.1, are rod shaped bacteria of about 2 μm in length and 0.5 μm in diameter. They are gram negative non sporulating bacilli. The presence of flagella makes them motile. *E. coli* is a facultative anaerobic, which means that, in the presence of oxygen they produce ATP by aerobic respiration, but in the absence of oxygen they are capable of switching to anaerobic respiration. These bacteria are commonly found in the lower intestine of mammals. Most *E. coli* strains are harmless although there are some pathogenic strains that may cause serious health issues, like food poisoning.

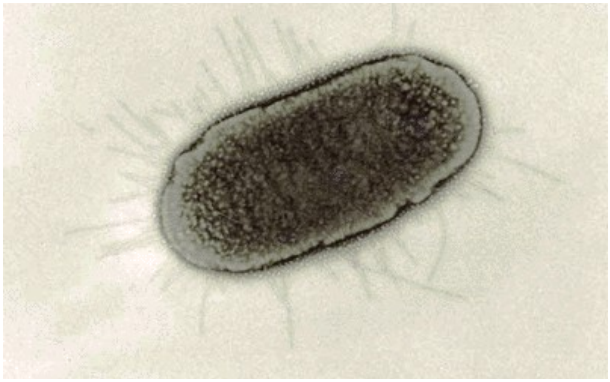


Figure 5.1: An image of a single *E. coli* bacterium.

E. coli plays an important role in maintaining intestinal physiology; the pathogenic strains can be classified on basis of differences in virulence attributes,

adherence patterns and their interaction with intestinal mucosa. Acid resistance is an important virulence property of *E. coli* and will be studied in Chapter 6. This is also observed in nonpathogenic strains. In order to colonize the mammalian intestine *E. coli* needs to break through the acid barrier of the stomach. In response to any kind of stress the bacterium has developed a particular mechanism of adaptation and has been the subject of intense research to investigate this at molecular and physiological levels.

It is a well known fact from previous studies [Neidhardt et al., 1990] that responses to both heat and cold in bacteria follow similar adaptive patterns. A sudden change in temperature brings variation of the expression profiles of a number of genes. On immediate temperature increase (also known as heat shock) *E. coli* and other bacteria induce the expression of protein chaperones and proteases[1], the role of which is to cope with heat-induced alterations. At the transcriptional level this variation can be controlled by two alternative sigma factors [2] *rphoH* and *rphoE*, known as σ^{32} and σ^{24} respectively. These specific promoters direct RNA polymerase to transcribe the implicated gene whether in cytoplasmic or extracytoplasmic function [Falciani, 2007].

Knowing the basic physiology of *E. coli* so far we are interested in finding the response and adaptation mechanism of our subject under temperature shift. To address this question microarray gene expression experiments were set up (as explained in Section 5.1.1). The gene expression profiles from microarray experiments were investigated following the pre-processing of data as described in Section 5.2. Post-analysis shows the application of our developed network analysis tool that captures the role of gobal transcription regulator FNR that controls the regulation of number of *E. coli* genes in response to change in temperature. In addition our network model predicts the activation of small heat and cold sensitive proteins in

[1]Chaperones facilitate protein folding and proteases break protein bonds.

[2]Sigma(σ) factors direct a specific RNA polymerase to transcribe a gene. Different sigma factors are activated in response to different environmental changes. However every RNA polymerase has a single sigma factor subunit. In *E. coli* there are seven different sigma factors.

adaption mechanism under temperature shift stress.

5.1.1 Expression profiling by microarray

In this case study we describe how *E. coli* K-12 responds during adaptation from $10°C$ to $37°C$. This process models the adaptation of *E. coli* in transit from an external environmental temperature to the human or animal host body temperature. The experiments in this case study and the data pre-processing steps described in Section 5.2 were carried out by Dr Francesco Falciani at the University of Birmingham.

Initially the system was characterized by determining the effect of the temperature shift on the growth rate. *E. coli* was grown for several days at $10°C$, including adequate aeration in Lysogeny broth (LB), which is a nutritious medium suited to *E. coli*. When reaching mid-exponential phase the temperature was increased to $37°C$ and the optical density was then measured over time (see Figure 5.2). Optical density(OD) is the measure of the amount of light absorbed by a population of bacterial cells in liquid suspension by use of a spectrometer or colourimeter. Figure 5.2 suggests that the temperature shifted culture enters a lag phase of about 10 minutes and once the lag is over, grows slower than the nonshifted temperature culture. This suggests that the adaptation to optimum temperature may be stressful for bacterial metabolism.

A microarray experiment was designed to address this hypothesis. This experiment compared the transcription profile of cells as they adapted from $10°C - 37°C$ with the transcription profile of a control culture growing at $37°C$. The control and experimental culture were at the same optical density at time $t = 0$. Considering $t = 0$ the time when the cells are shifted from $10°C - 37°C$. The changes in the transcription profiles were monitored from these two cultures for 13 time points using a single channel glass microarray that represented the entire genome of the K-12 strain. Three experimental replicates were performed using labelled cDNA.

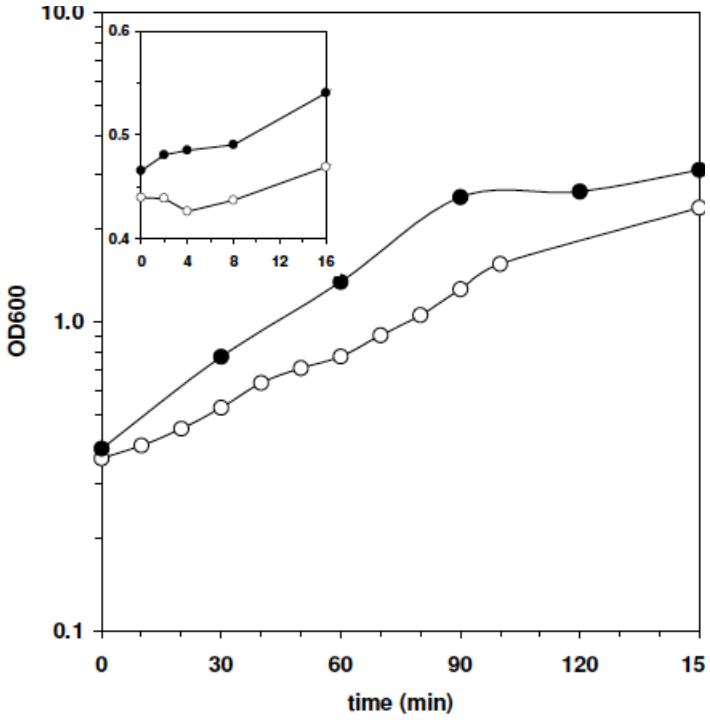


Figure 5.2: The growth curve of control and temperature shifted *E. coli* $MG1655$ bacterial cells. The vertical axis represents OD whereas the horizontal axis represents time. The curves with closed circles represent the growth of the control strain at $37°C$ and the curves with open circles represent the growth of the strain that undergoes temperature shift from $10°C$ to $37°C$. The inset shows the early sampling from time $0 - 15$ mins of the culture in closely spaced time points [Falciani, 2007].(This particular figure is provided by Dr Francesco Falciani as a part of a biological experiment result.)

This was hybridized on three different slides for each time point in order to consider potential experimental and technical variability. Each experimental replicate had two technical replicates. Under each condition the single channel hybridization was carried out and 4290 gene expression measurements were taken for each of the 13 time points.

5.2 Data Preprocessing

A microarray dataset $Y_{i,j}$ is a set of real numbers, where $i = 1, 2, \cdots, n$ corresponds to the probes on the arrays, representing n genes and $j = 1, \ldots, m$ stands for the

sample size m, for e.g., $m = 2$ for the two color glass chip technology. The values of $Y_{i,j}$ with $i = 1, \ldots, n$ and $j = 1, \ldots, m$ are the intensity data that is produced by the image quantization software.

Any experimental measurements in the context of gene expression profiling are subject to systematic errors. These errors occur through natural biological variation or through technical means. The process of removing these errors is called normalization. These errors need to be addressed before proceeding with further data analysis. In this section we describe in detail the chosen method for normalization and the quantification of differentially expressed genes.

5.2.1 Variance Stabilization Normalization (VSN)

The data were normalized using the VSN [Huber et al., 2002] method, which is a parametric model assuming that the different samples can be represented on the same scale based on linear mappings. The proposed method is based on variance stabilization, which is used to derive a transformation h such that the variance $var(h(Y_{i,j}))$ is independent of the mean $E(h(Y_{i,j}))$. The transformation h takes the parametric form $h(x) = \text{arcsinh}(a + bx)$ and is derived from a model of the variance-versus-mean dependence. $\hat{Y}_{i,j} = \boldsymbol{Y}_i$ is regarded as a random variable with mean $E(Y_i) = \mu_i$ and variance $var(Y_i) = v_i$. Here v_i only depends on i through a quadratic function of the mean μ_i.

$$v_i = v(\mu_i) = (c_1\mu_i + c_2)^2 + c_3,$$

where $c_3 > 0$. The method of variance stabilization can then be used to derive a transformation h such that the variance $var(h(Y_i))$ is approximately independent of the mean $E(h(Y_i))$. Following the study by Tibshirani [1988] on variance stabilization, h can be given as

$$h(y) = \int_0^y 1/\sqrt{(v(u))}du,$$

where y represents the element of Y_i and results from a linear approximation of $h(Y_i)$ around $h(\mu_i)$. Substituting equation for $v(\mu_i)$ into $h(y)$ we get

$$h(y) = \gamma \operatorname{arcsinh}(a + by),$$

where the parameters of h are given by $\gamma = c_1^{-1}$, $a = c_2/\sqrt{c_3}$ and $b = c_1/\sqrt{c_3}$. The relationship between the arcsinh function and the logarithm can be shown as

$$\operatorname{arcsinh}(x) = \log(x + \sqrt{x^2 + 1}),$$

$$\lim_{x \to \infty} (\operatorname{arcsinh}(x) - \log(x) - \log(2)) = 0.$$

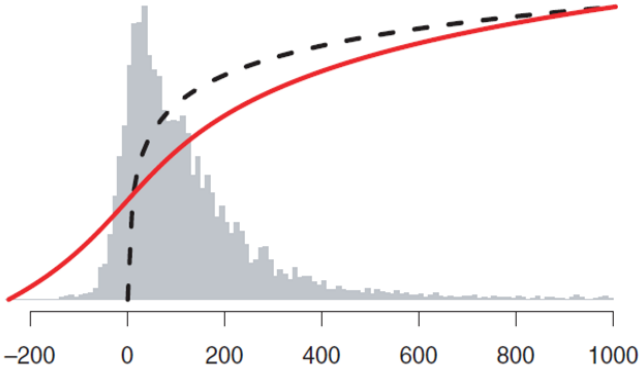


Figure 5.3: From Huber et al. [2002]: This graph represents the variance stabilizing transformation using the arcsinh function (solid line) and the logarithm function (dashed line). For the temperature shift dataset the variance stabilizing transformation uses the arcsinh function. The histogram shows the gene intensity distribution.

This means that for a larger number of probes the arcsinh transformation is the same as logarithmic transformation. The advantage of this transformation against log transform is that it does not have a singularity at point zero as shown in Figure 5.3. Moreover it continues to stay smoother and real-valued in the range of small or negative intensities. Through the Bioconductor project the author also provides his method as an R package that is publicly available. The temperature

shift dataset was normalized using the Bioconductor implementation.

5.3 Detecting differentially expressed genes

Stegle et al. [2010] proposed a Gaussian process (GP) based two sample test (GP2S) that detects differentially expressed genes. This was applied to *E. coli* gene expression levels from three biological samples that are exposed to two different conditions i.e. under control ($37°C$) and temperature shift ($10°C$ to $37°C$) conditions. The goal here is to determine whether a given gene probe is differentially expressed between these conditions.

The first model in the GP2S test assumes that the microarray time series in both conditions control (C) and temperature shift (S) are samples drawn from an identical shared distribution $\mathbf{f}$. The other model describes the time series in both conditions as sampled from two independent distributions, $(\mathbf{f}_C(t), \mathbf{f}_S(t))$. Firstly, in the shared model (H_s) the joint posterior distribution is taken over the unobserved function value $\mathbf{f}$ and the replicate observations $Y_{i,j}^{(C,S)}$ for conditions (C,S). The covariance function used decays exponentially with squared distance of time (i.e. $k_{SE}(x) = A\ exp\frac{1}{2}\frac{x^2}{l^2}$). This provides a function with parameters of amplitude (A) and length-scale, l (known as kernel hyperparameters, θ_K).

Secondly for the alternative hypothesis (H_I) the posterior distribution is defined by taking the independent product of Gaussians. The hyperparameters of this independent model are optimised jointly for both the control and shift processes $\mathbf{f}_C(t)$ and $\mathbf{f}_S(t)$ respectively, where kernel hyperparameters θ_K and the global noise variance σ are shared. In this way the number of hyperparameters remains the same in both models. The two alternatives, the shared model (H_S) and the independent model (H_I) can then be objectively compared using the following score

$$Score = \log \frac{P(D_A, D_B|H_I)}{P(D_A, D_B|H_S)}, \tag{5.1}$$

where D_A and D_B are observed expression levels in two conditions A and B. A typical result obtained from GP2S is shown in Figure 5.4.

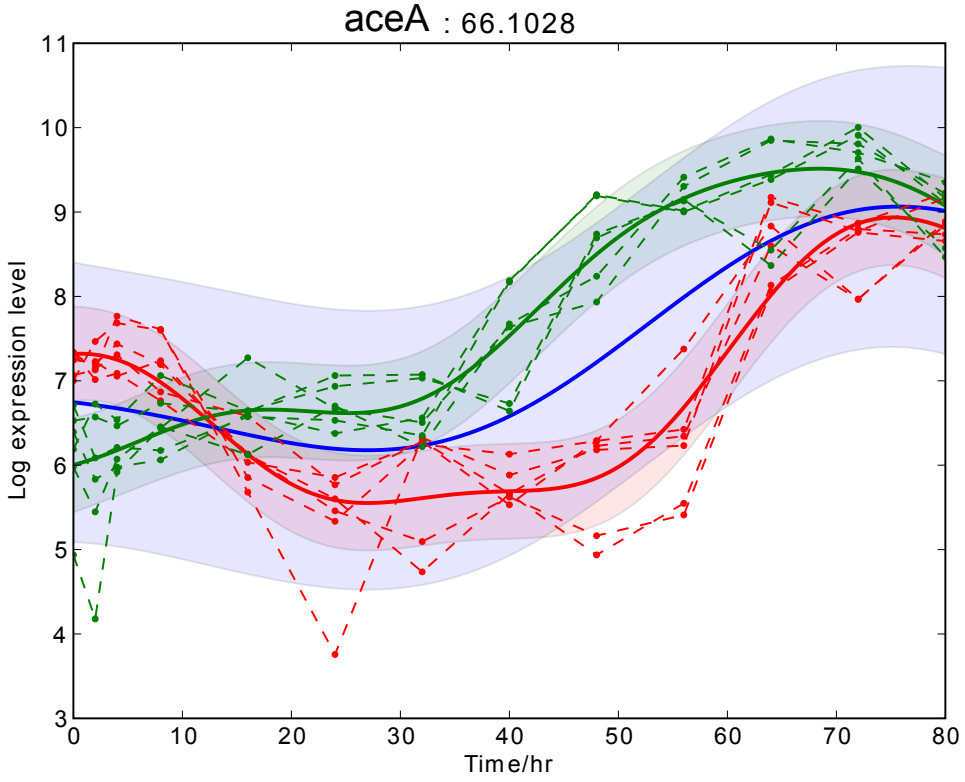


Figure 5.4: An example result produced by the GP2S test. Dashed lines represent replicates of gene expression measurements for control (green) and temperature shift (red). Thick solid lines represent Gaussian process mean predictions of the latent process traces; ±2 standard deviation error bars are indicated by shaded areas. The value on top of the plot represents the score according to equation 5.1.

5.4 Data exploration

5.4.1 Clustering

Data clustering is a method that groups together objects that are similar in selected characteristics. The goal of clustering is to reduce (or to rearrange) the amount of data by categorizing (or grouping) similar items together. Clustering methods can

generally be divided into two categories: hierarchical and partitional. Within each of these types there exist many different algorithms for defining the clusters. Two approaches are top-down and bottom-up. In a top-down approach larger clusters are split into smaller clusters and in a bottom-up approach each datum is considered as an individual cluster, and smaller clusters are merged into larger ones. The clustering methods differ in the rules by which it is decided which two smaller groups are merged, or which larger cluster splits. However, in the end a dendrogram shows how the clustered items are related.

Partitional clustering attempts to decompose the data set into a set of disjoint clusters. K-means clustering is a commonly used partitional clustering method. Implementation of this algorithm is easier but there are some disadvantages such as the interpretation of the clusters and the choice of the number of clusters.

For the analysis of microarray time series from the temperature shift dataset we use the Bayesian hierarchical clustering(BHC) approach of Savage et al. [2009]. The advantages of this method are that it does not require us to define the number of clusters in advance, and it uses a probabilistic method of deciding which clusters to merge. The BHC algorithm is based on a fast Dirichlet process (DP) clustering method where an infinite mixture model is used to model the data and Bayesian model selection is used to decide when the clusters should be merged. This bottom-up method starts with initializing each data point in its own cluster and then it merges pairs of clusters iteratively. In order to merge two clusters this method uses a hypothesis test. The first hypothesis is that data (D) were generated independently and identically from the same probabilistic model, $p(x|\theta)$. The alternate hypothesis states that there are two or more (than two) clusters in D. The marginal probability of the data is defined by combining the probability of the data under each hypothesis. The posterior probability of the merged hypothesis M was obtained using the Bayes' factor. The rule was then set, if $M > 0.5$ indicates merged hypothesis to be more probable than alternative partitioning one and if $M < 0.5$ then the cluster braches

remains as seperate clusters.

After detecting differentially expressed genes from GP2S, the list of genes was sorted using corresponding Bayes factor based on equation 5.1. For clustering of the temperature shift dataset we have used an extension of the BHC algorithm that is suitable for time series microarray data and based on a GP likelihood function [Cooke et al., 2011]. One of the special features of this algorithm is that it can take outlier measurements into account. Kuss et al. [2005] described the use of a mixture likelihood of a two-model, with outliers (p_o) and without outliers (p_r) in order to obtain a robust Bayesian regression model that incorporates outliers in observations. Assuming the existence of an outlier and believing its distribution to be different from p_o, then by using the fraction of outliers π we can combine both models p_o and p_r as

$$p(Y_{i,j}|\mathbf{f},\theta) = (1-\pi)p_r(Y_{i,j}|\mathbf{f},\theta) + \pi p_o(Y_{i,j}|\mathbf{f},\theta) \tag{5.2}$$

Where $\mathbf{f}$ is the latent function of a GPR model, θ is a set of hyperparameters and p_o, p_r are Gaussian density functions with different variance.

This results in a smaller number of clusters than when outlier measurements are ignored. For the GP likelihood function this algorithm provides two choices of a covariance function i.e. a squared exponential and a cubic spline. For this case study we have used BHC with both covariance functions and with and without incorporating outlier measurements. We obtained the minimal number of clusters with a combination of a squared exponential covariance function with outlier measurements. Figure 5.5 shows the heatmap output by BHC for the top 1400 differentially expressed genes from GP2S for the temperature shift data. In the heatmap the differential expression increases from underexpression (i.e. green) towards over expression (i.e. red). On the left hand side there is a dendrogram that shows the merging of the clusters. The blue lines represent the branches of accepted clusters while the red dashed lines are merges that were rejected by the algorithm.

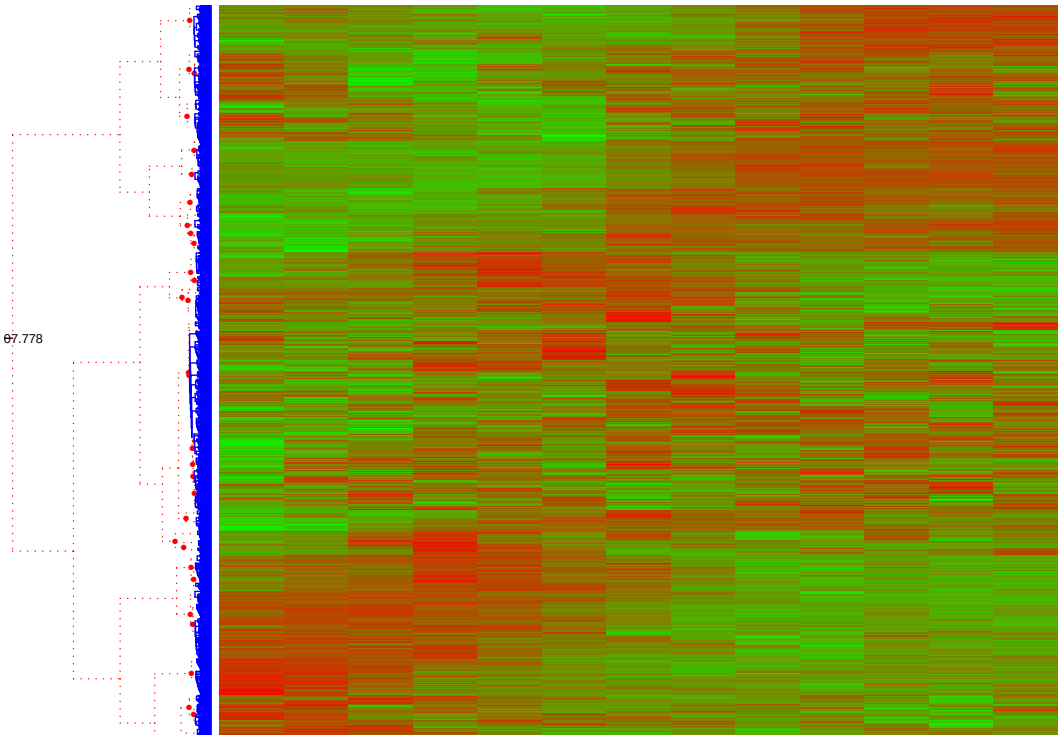


Figure 5.5: Heatmap representation of clustering of the top 1400 differentially expressed genes for the temperature shift dataset. The dendrogram is shown on the left of the heatmap. The blue lines show accepted merges of clusters. The red dotted lines represent the merges rejected by the algorithm.

For the validation of the clustering procedure we investigated whether genes from an operon are within a cluster. For an overall 1400 genes (these genes are on microarray) there are 467 operons with 613 genes [Huerta et al., 1998]. Our task here is to check how many genes from an operon are found intact in a cluster and how many times genes from an operon splits between cluster. For this we made following contingency table.

For example the operon *aroL-yaiA-aroM* which includes genes *aroL, aroM* and *yaiA* appears in cluster 14. Similarly the operon *malXY* which includes genes *malX* and *malY* appears in cluster 16. More operons are given in Table 5.1. Overall, 15 clusters out of 52 contained complete sets of multiple genes from an operon.

No. of Cluster	operon ID	list of genes involved in an operon
14	aroL-yaiA-aroM	aroL , aroM, yaiA
16	malXY	malX, malY
18	dmsABC	dmsA, dmsB, dmsC
21	cadBA	cadA, cadB
22	cysJIH	cysH, cysI , cysJ
22	cysDNC	cysC, cysD, cysN
27	rpmH-rnpA	rnpA, rpmH
27	yjiXA	yjiA , yjiX
32	glpABC	glpA, glpB , glpC
33	narGHJI	narG, narH, narI , narJ
37	carAB	carA , carB
42	sdaCB	sdaB, sdaC
42	dadAX	dadA, dadX
42	cyoABCDE	cyoA, cyoB, cyoC, cyoD, cyoE
51	aceBAK	aceA, aceB, aceK

Table 5.1: Clusters that contain genes comprising an operon.

	in same cluster	not in same cluster	
Genes in Operons	423	190	613
Genes not in Operons	787	0	787
	1210	190	1400

Table 5.2: Contingency table that summarises 1400 genes and the corresponding number of operons. Applying Fisher test to check our null hypothesis that operon genes are independent of clustered genes, results in a p-value of $2.57e-77$ (i.e. < 0.05), therefore indicating there would be a statistically significant association between the operon genes found in a cluster and actual operons.

5.4.2 Eigengene analysis

The literature about genes and their protein products provides evidence that they are organized into functional clusters such as molecular, biological, cellular processes and pathways. With the help of available literature such as Huerta et al. [1998], Salgado et al. [2004], Keseler et al. [2005], Salgado et al. [2006] and online databases like *RegulonDB* and *EcoCyc* we can find information about how to detect biologically significant and meaningful clusters in networks. However there is a need for an appropriate method that allows us to study the relationships between clusters. From

the BHC clustering results we have observed that sets of genes that were grouped together share similar ontology (explained further in Section 5.5). We seek to reduce the dimensionality of the data by using a representative summary of the shape of each gene expression cluster.

Langfelder and Horvath [2007] have proposed a method that can be used to describe the relationships between co-expression clusters. Firstly they use a method that detects clusters that are shared by two or more networks and then represent gene expression profiles by an eigengene (EG). This study further leads to EG networks which give a global view with an effective and biologically sound way of representing relationships between clusters of a gene.

In this section we describe how to represent a cluster by using Singular Value Decomposition (SVD) of the clustered expression values. After clustering we assume that the gene expression values of the n^{th} cluster are denoted by $G^{(n)} = (g_{ij}^{(n)})$, where the index $i = 1, 2, \ldots, k$ represents the clustered genes and the index $j = 1, 2, \ldots, m$ represents the time points. It is assumed that the data is normalized to a mean of zero and a variance of 1. The singular value decomposition of $G^{(n)}$ (illustrated in Figure 5.6) is then denoted by

$$G^{(n)} = UDV^T$$

Variables U and V are orthogonal matrices of dimension $U^{(n)} = k^{(n)} \times m$ and $V^{(n)} = m \times m$. Variable D is a diagonal matrix of dimension $m \times m$. The first column of the $V^{(n)}$ is defined to be a cluster eigengene [Langfelder and Horvath, 2007].

Figure 5.7 is an example of an eigengene obtained from the first cluster of the temperature shift dataset. The top plot shows gene profiles that are clustered together from the set of the top 1400 differentially expressed genes. The middle

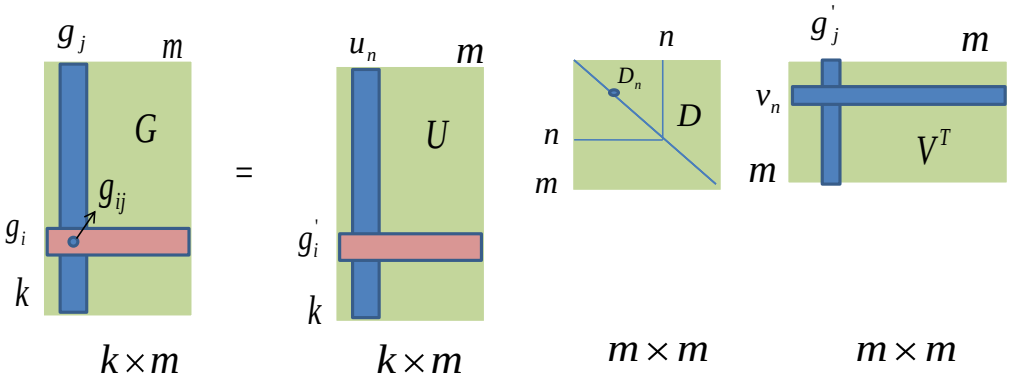


Figure 5.6: Singular value decomposition of G matrix into U, D, and V from Wall et al. [2003].

plot is the eigengene profile of this cluster. The bottom plot combines the clustered profiles of cluster genes (in red) and the eigengene (in thick black). Calculated EGs from this section will be used for network inference (as described in the following Section 5.5)

5.5 Functional annotation of the genes

In order to test the hypothesis that the clusters we identified represent a coordinated functional response we performed a functional analysis of each clustered gene. This task has been achieved by performing GO enrichment analyis as described in the section 5.5.1.

5.5.1 Gene Ontology

The concept of ontologies is widely used to create a controlled set of vocabulary that communicates and annotates knowledge. The Gene Ontology is an international standard to annotate genes. The Gene Ontology (GO) is a bank of biological terms that gives insight into the functional characteristics of genes. Each GO term has zero or more annotations arranged in a tree where parent terms inherit annotation

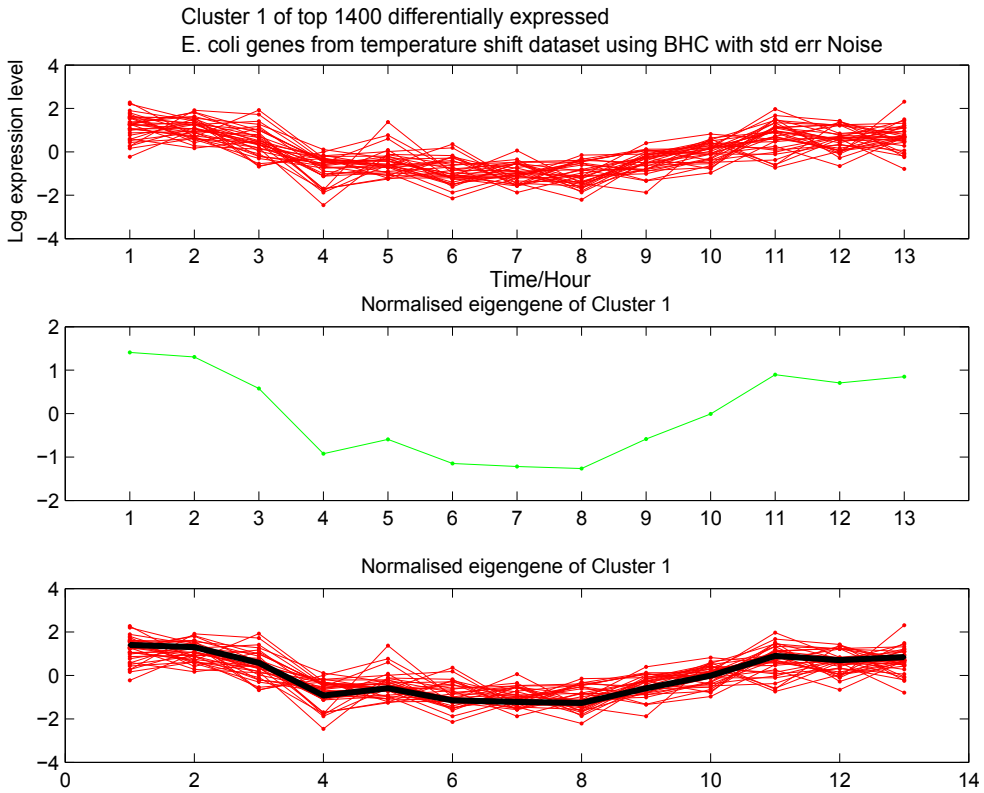


Figure 5.7: Eigengene from the first cluster of genes from the temperature shift dataset.

from children as shown in Figure 5.8. Each category such as biological process (BP), molecular function (MF) and cellular component (CC) are further divided into more specific detailed processes such as transcription factor activity $\longrightarrow$ transcriptional regulator activity.

5.5.2 Hypergeometric test

The basic argument behind this approach is that there is a universe of genes that can be divided into two groups i.e. those that are of interest and those that are not. In addition there are other characteristics of these genes such as belonging to certain GO categories, such as BP, MF, CC or having particular biological properties. To

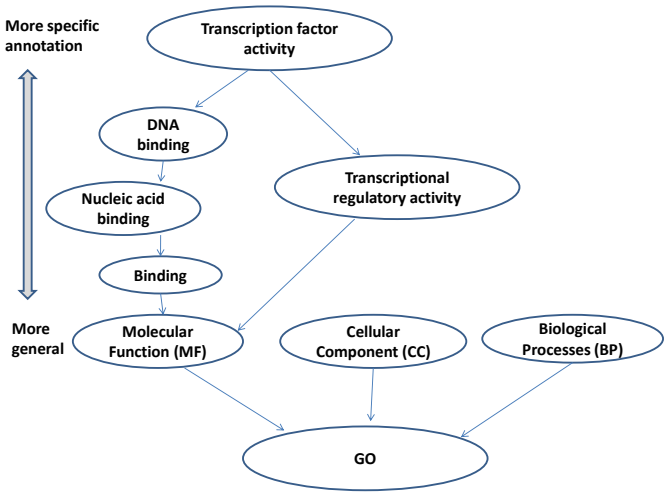


Figure 5.8: GO diagram: Edges go from children to parents showing that downstream parents inherit annotation from children.

identify the association between being interesting and having the particular property, a basic statistical test known as Fisher′s exact test is used.

The Gostats package in R provides the facility to test over- and under-representation of gene sets by using the hypergeometric test. The hypergeometric distribution is often used to model the number of successes in a sequence of n experimental draws without replacement from a population size of N.

The overrepresentation of GO terms from the set of the GO universe can be considered as an urn model. Here the urn is the universe of GO terms containing interesting genes coloured black while not interesting ones are coloured white. If there are j interesting genes in the GO category it is possible to compute the probability of seeing j genes in K draws without replacement. Moreover there is no reason why the grouping needs to be binary. There could be three types of genes (very interesting, interesting, and not interesting), and a category that has three or more levels. If so then the hypergeometric test needs to be generalised to address multivariate problems.

When using the *hypertest* it is appropriate to include only those genes in the universe that are of interest or are selected. Selection of the universe is crucial; if it is

too large or too small this will have a large impact on the observed p-values. Another practical issue to consider when analysing data from microarray experiments is that more than one probe on the array represents more than one gene. The bias might cause a problem for the hypergeometric test to be correct. For this reason it is good to use unique gene identities.

The output of the *hypertest* can be summarized by returning the results for terms that have a p-value less than a specified cut-off. Figure 5.9 represents one way of visualising the resulting annotation matrix. The annotation matrix is a heatmap representation of all genes on the x-axis. In this figure gray and white colour strips are used to distinguish sets of clusters. The black bars in the middle of the plot flags the overrepresentation of a particular annotation at a p-value threshold of ≤ 0.05.

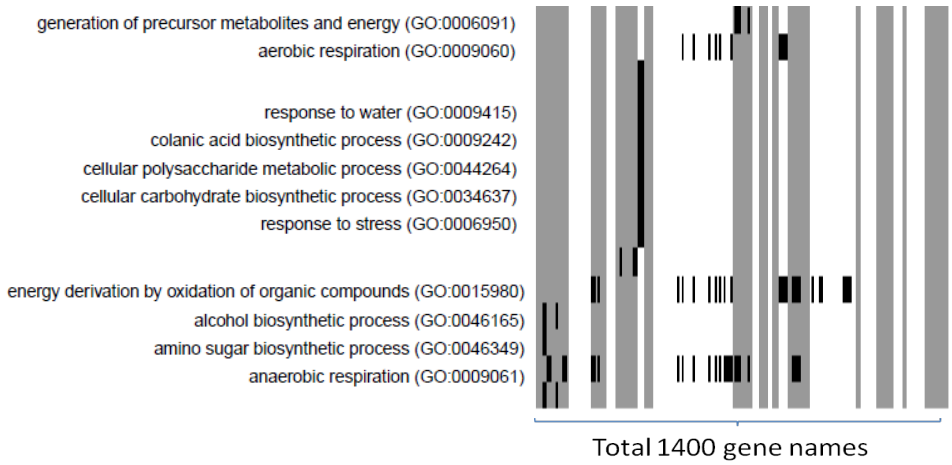


Figure 5.9: Annotation matrix: GO terms are specified on the left, while the x-axis represents genes. Black bars in the middle of the plot flag the presence of annotation.

5.5.3 Interpretation of the GO analysis

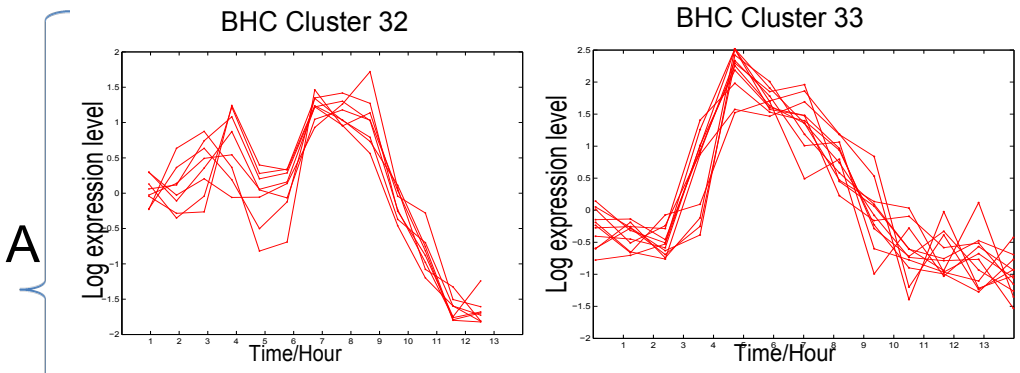
In the following part of this section we discuss the two examples of cluster annotation that involve anaerobic and aerobic respiration. They are representative of the results we describe in section 5.6.2.

Anaerobic respiration is a respiratory process that takes place in the absence of oxygen by using electron acceptors through an electron transport chain. In this processes chemical energy is converted to an electrochemical gradient and is used by ATP synthetases to produce ATP. This respiration plays a key role in the nitrogen, carbon and sulphur cycles. Figure 5.10 shows the list of genes and corresponding clustered profiles obtained from the clustering analysis described in Section 5.4.1, with statistically significant over representation of terms involved in anaerobic respiration, and with marker genes in red.

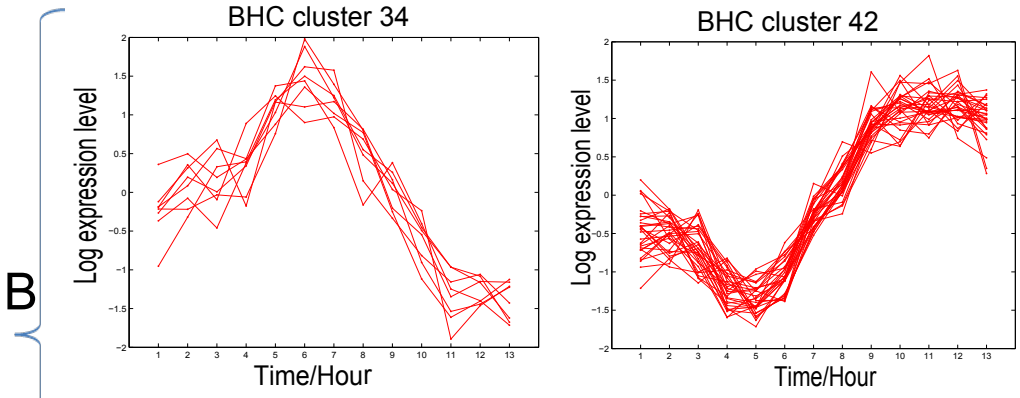
Aerobic respiration is a process that uses oxygen in order to break down molecules, producing energy by releasing electrons. It is also known as cellular respiration. This process creates the energy molecule ATP. Figure 5.10 shows two clusters from the clustering analysis of the data containing genes that are involved in the aerobic respiration process.

Panel (A) shows the profiles of the clustered genes that are involved in anaerobic respiration from clusters 32 and 33. It is interesting to find the two operons of *glp* and *nar* which are known to be involved in anaerobic respiration. Panel (B) shows the profiles of the clustered genes which are involved in aerobic respiration from clusters 34 and 42. The figure also describes the detailed annotation of these genes, for example the *hya* operon representing aerobic respiration activity. Similar activity was observed in the cluster 42 among *purR, secA, cyoA* genes.

Marker genes are a highlighted set of genes or operons that are indicators of special biological functions. For example in response to high temperature, *E. coli* bacteria cells activate a series of two component systems and transcription factors that are special regulatory factors involved in temperature adaptation. Some genes are well studied and can be found from literature or an open source such as *RegulonDB*.



# cluster	Unique Gene ID	GO term	BP/MP/CC	Marker Genes
32	GO:0006071	glycerol metabolic process	BP	glpABCK
32	GO:0009061	anaerobic respiration	BP	
33	GO:0015980	Anaerobic respiration	BP	narGHIJ



# cluster	Unique Gene ID	GO term	BP/MP/CC	Marker Genes
34	GO:0009061	aerobic respiration	BP	hyaBCDE
34	GO:0015980	energy derivation by oxidation	BP	yhhM
42	GO:0016564	transcription repressor activity	MF	dadA
42	GO:0009060	aerobic respiration	BP	purR,secA,cyoA

Figure 5.10: Panel A shows an example of anaerobic respiration from clusters 32 and 33, with up-regulated genes from operon *glp* and *nar*. Panel B shows an example of aerobic respiration from clusters 34 and 42, with up-regulated genes such as *purR*, *secA*, *cyoA*.

5.5.4 Functional Anannotation Clustering

After gene enrichment analysis using the hypergeometric test we would like to measure the relationship among the annotation terms. In this case a group of genes from a cluster were further investigated using a new feature of the DAVID Functional Annotation Clustering tool (mainly web based) [Huang et al., 2009a], [Huang et al., 2009b]. DAVID applies a novel algorithm (that integrates the Kappa statistics and fuzzy heuristic clustering) to cluster functional terms that contain a large proportion of genes in common in a smaller number of functional clusters.

Kappa statistics measure the degree of similarity between common genes between two annotations and fuzzy heuristic clustering classifies the groups of similar annotations as per the kappa values. In this way the more commonly shared gene annotations get a higher chance to group together. The annotation results are followed by the p-value (Fisher Exact test) associated with each annotation term and the Benjamini false discovery rate for the BHC clusters.

Table 5.3: The table shows a representative term for clusters. This is chosen for the most significent terms. Gene annotations for each cluster are sub-divided into three clusters with the term of highest significance shown. The column named "Clust ID" indicates the cluster number. In the column "Anno clust" the annotation clustering is given. "GO-BP" are gene ontology biological processes, "INT" are INTERPRO based annotations, "KEGG" are Kyoto Encyclopedia of Genes and Genomes based annotations. The column with "Representative Terms" specifies the functionality of gene, followed by p-value and Benjamini false discovery rate.

Clust ID	Anno Clust.	ES		Representative Term	P˙value	FDR
1	1	3.45	GO˙BP	amine biosynthetic process	1.1E-05	1.5E-03
	3	1.27	KEY	signal	8.3E-03	1.3E-01
7	1	1.83	KEY	cell membrane process	5.1E-05	3.0E-03

Continued on next page

Table 5.3 – *Continued from previous page*

Clust ID	Anno Clust.	ES		Representative Term	P'value	FDR
	3	1.15	KEY	4fe-4s	5.4E-03	7.8E-02
8	1	1.79	KEY	oxidoreductase	1.3E-03	6.1E-02
9	1	1.27	KEY	cell inner membrane	1.3E-03	6.9E-02
10	1	2.78	KEGG	Pyruvate metabolism	1.9E-03	4.3E-02
	2	1.76	KEY	amino-acid biosynthesis	4.1E-03	4.0E-02
	3	1.75	KEY	oxidoreductase	1.7E-03	2.4E-02
12	1	0.86	KEY	transcription regulation	7.8E-02	6.0E-02
13	1	1.66	KEY	oxidoreductase	7.1E-04	1.3E-02
	2	0.81	KEY	membrane	1.1E-02	9.6E-02
14	1	2.18	KEY	amino-acid biosynthesis	6.3E-04	8.6E-03
	2	2.15	INTE	Cold shock protein	1.1E-05	8.4E-04
	3	1.42	KEY	cell membrane	2.9E-04	7.9E-03
15	1	2.36	KEY	metal-binding	1.7E-04	1.1E-02
17	1	2.01	KEY	hydrolase	2.7E-04	1.3E-02
18	1	2.07	GO-CC	organelle envelope	2.1E-03	4.2E-02
	2	1.86	KEY	cell membrane	7.9E-05	4.3E-03
	3	1.85	GO-BP	organic acid transport	4.5E-02	5.1E-03
19						
20	1	2.19	GO-MF	glucosidase activity	1.6E-04	8.8E-03
0	2	1.19	KEY	transcription regulation	1.1E-02	9.1E-02
22	1	12.02	GO-BP	sulfate assimilation	5.1E-14	1.2E-12
	2	3	KEGG	Sulfur metabolism	1.6E-10	4.8E-09
	3	2.97	KEY	Cysteine biosynthesis	7.2E-12	3.0E-10
24	1	3.94	INT	RNA polymerase	1.8E-13	7.7E-12

Continued on next page

Table 5.3 – *Continued from previous page*

Clust ID	Anno Clust.	ES		Representative Term	P'value	FDR
				-binding, DksA		
25	1	1.99	KEY	transmembrane protein	8.5E-04	3.1E-02
	2	1.15	KEY	metal-binding	4.4E-03	4.0E-02
				biosynthetic process		
26	1	2.47	KEGG	Lipopolysaccharide	2.1E-04	2.0E-02
				biosynthetic		
29	1	1.77	KEY	transport	1.5E-04	3.2E-03
	2	1.76	KEY	electron transport	1.4E-05	6.0E-04
30	1	3.27	GO-BP	anaerobic respiration	1.0E-04	1.3E-02
	2	2.61	GO-BP	cellular carbohydrate	7.7E-04	2.5E-02
				biosynthetic process		
	3	2.17	KEY	metal-binding	1.4E-05	1.2E-03
32	1	3.64	GO-BP	glycerol metabolic process	1.9E-11	2.5E-10
	2	2.66	GO-MF	glycerol-3-phosphate	7.0E-05	1.2E-03
				dehydrogenase activity		
33	1	5.09	GO-MF	nitrate reductase activity	4.5E-09	1.3E-07
	2	4.16	KEY	oxidoreductase	5.2E-09	7.1E-08
34	1	4.62	GO-BP	aerobic respiration	9.9E-06	1.6E-04
	2	2.7	KEY	metal-binding	7.5E-07	2.3E-05
35	1	2.98	KEGG	Fructose and mannose	1.1E-05	2.5E-03
				metabolism		
	2	1.3	KEY	cell inner membrane	8.3E-05	2.4E-03
	3	1.13	KEY	oxidoreductase	4.2E-04	9.1E-03
38	1	7.45	KEY	flagellum	1.5E-18	2.0E-17

Continued on next page

Table 5.3 – *Continued from previous page*

Clust ID	Anno Clust.	ES		Representative Term	P'value	FDR
	2	5.87	KEGG	Flagellar assembly	1.3E-06	1.6E-05
	3	3.43	GO-MF	maltose transmembrane transporter activity	5.8E-05	1.0E-03
39	1	1.1	KEY	cell inner membrane	4.8E-03	8.9E-02
41	1	1.97	KEY	transport	3.6E-05	3.1E-03
42	1	4.67	KEGG	Oxidative phosphorylation	7.7E-08	2.2E-05
	2	3.51	INTE	Alanine racemase region	2.3E-05	1.5E-03
	3	2.04	INTE	Serine dehydratase beta chain	4.7E-04	1.0E-02

5.6 Inference of gene regulatory network

In this section our objective is to make use of the clustered genes and infer the regulatory network using the MCMC based algorithm as developed and discussed in the earlier chapters. The informative clusters from the inferred network structure might provide useful hypotheses that could address the function of genes involved in the key role of stress response.

5.6.1 Computational experiment

So far most GRN inference studies have been limited to the analysis of smaller numbers of genes. To overcome the challenges of modeling large numbers of genes we have followed the approach proposed by Hirose et al. [2008], to infer gene regulatory networks by making use of the putative transcriptional clusters in which genes share a common expression profile. As described in Section 5.4.2 we collected a dataset of 52 EGs from the clustering results of the BHC algorithm using the top 1400

differentially expressed genes. Each EG represents a potential transcriptional cluster where clustered genes are highly correlated in expression either because they are operon or because they are involved in the same pathways. Each EG was calculated from 13 time points and 6 replicates (considering that each of the three biological replicates has two technical replicates).

Using the set of EGs as "pseudo genes" for the network inference, the sampler was set to run for $150,000$ iterations from five randomly chosen starting points. Every 10^{th} drawn sample was then saved from each chain. The hyperparameters were updated after every 1000^{th} iteration where the Metropolis-Hastings iterations within Gibbs were set to run for at least 5000 iterations. This numerical experimental set$-$up was then repeated for different SSM models by increasing hidden state space dimension $k = 1, 2, \cdots, 10$. The Gibbs sampler output from all the model parameters and the hidden states were collected and further investigated by measuring the convergence of the Markov chains using the PSRF calculations (as described in Section 2.3.1). Figure 5.11 summarises the results from the Gibbs sampler algorithm. Panels (i), (ii), (iii), (iv) show the plots of calculated PSRF values for the dynamic parameters of the model $(\mathbf{A}, \mathbf{D})$ and for the noise parameters $(\mathbf{Q}, \mathbf{R})$. The PSRF value calculated for $(\mathbf{B}, \mathbf{C})$ are similar to $\mathbf{A}$. The Y-axis of these subplots represents the PSRF value. The X-axis shows the increasing size of the parameter matrix with respect to the increasing dimension of the hidden state space. Different colors in the subplots help to distinguish the increasing dimension of hidden state space dimension. As shown most of the parameters are fluctuating below the acceptance margin indicated by the straight line at 1.1. After observing convergence from the PSRF values for each of the MCMC chains our next step is to calculate the model evidence by using Chibb′s algorithm (described in Chapter 2).

In Figure 5.11 (panel v) for each hidden state space dimension the error bar is calculated from at least 5 model evidence values resulting from different independent Markov chains. The marginal likelihood calculated from the variational

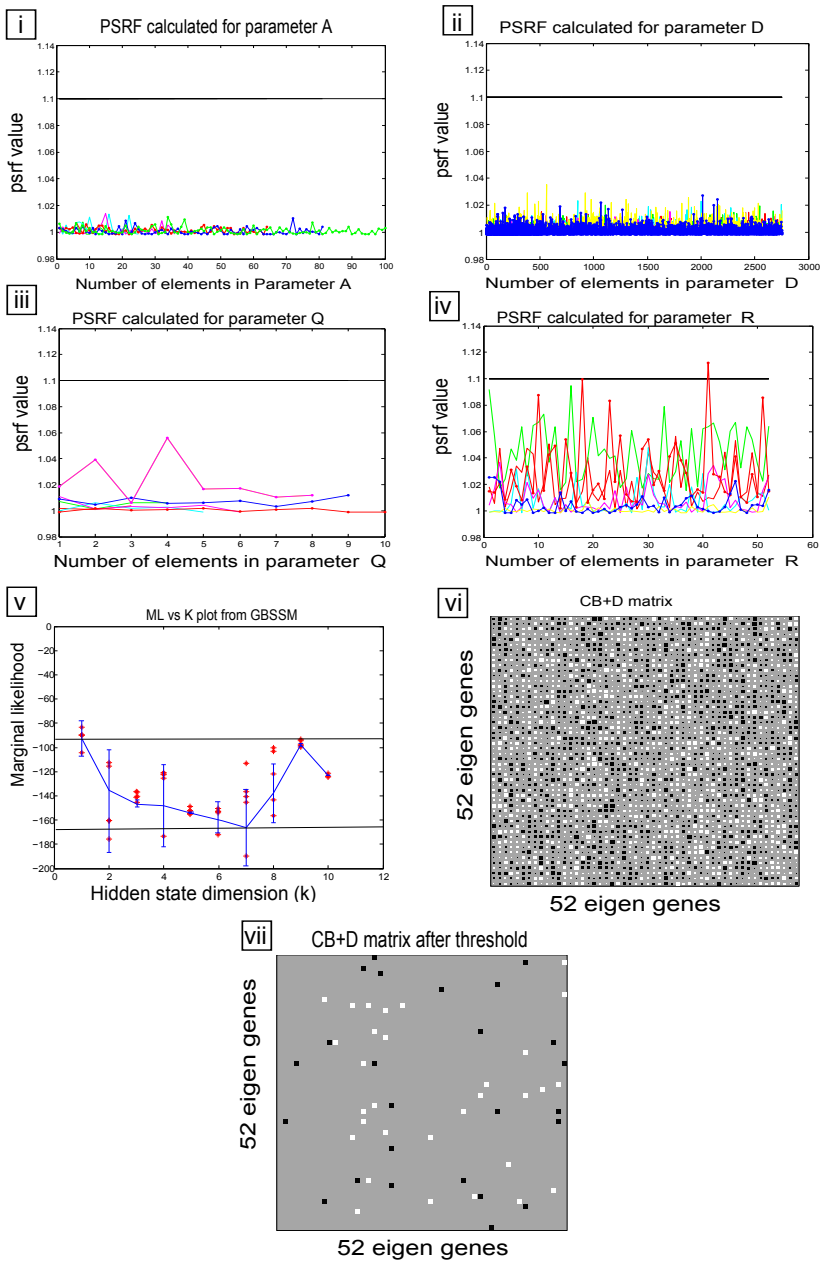


Figure 5.11: Summary of MCMC output results. Panels (i,ii) of this figure represent the PSRF calculation for the dynamic parameters $\mathbf{A}, \mathbf{D}$ and panel (iii,iv) represents the PSRF calculated for the noise parameters $\mathbf{Q}, \mathbf{R}$. Panel (v) shows the plot of model evidence versus hidden state dimension. The model evidence was calculated using Chibb's method and the hidden dimension $k = 1$ gives the maximum marginal likelihood (some evidence of observability at $k = 9$). The Hinton diagram in panel vi shows the gene-gene interaction matrix which was estimated from an average over the 5 Markov chains using the corresponding dynamic parameters from the model $k = 1$. Panel (vii) shows the Hinton diagram after thinning the interaction matrix by using the 95% confidence interval.

Bayesian approach is the lower bound to the ML plot and is indicated by a straight line. Moreover the upper bound was chosen on the basis of the highest, averaged marginal likelihood, calculated for different SSMs. This is to show that the marginal likelihood calculated from different hidden state space dimensions lies within the bound. We can observe a change in the trend of the estimated ML after the hidden state space dimension $k = 8$, possibly indicating the overfitting of the parameters of the SSMs. For this reason, we discard results from $k \geq 8$ and find the maximum value of the model evidence from $k = 1, \cdots, 7$. It can be observed that the marginal likelihood attains a maximum value at $k = 1$. Therefore by using the estimated parameters from the hidden state space dimension of $k = 1$ and the average over 5 independent Markov chains, we have calculated the gene-gene interaction matrix $[\mathbf{CB} + \mathbf{D}]$ (as described in Chapter (4) 4 Section (4.3.2) 4.4). The Hinton diagram in Figure 5.11 panel VI represents the calculated gene-gene interaction values. The interaction matrix is then thresholded by using the Z statistics with a confidence interval of 95%. The Hinton matrix in Figure 5.11 panel VII represents these thresholded interactions. Using the Cytoscape tool [Smoot et al., 2011] the inferred gene regulatory network can be presented as in Figure 5.12.

5.6.2 Results and discussion of the inferred network

In the inferred network shown in Figure 5.12 each node represents a set of clustered genes (represented by that EG). Every EG has its specific functional annotation (description) and by following careful investigation we can link the regulatory network information to the significant biological processes. Potentially interesting biological processes are highlighted in colored nodes and their corresponding functions are given in Table 5.4. However further investigation is required to address our main interest of understanding the resistance mechanism of bacterial cells under the condition of temperature shift.

Consistent with our understanding of stress response, most of transcription

factors (i.e. 11 out of 52) are found in clusters at the top level of the hierarchy (shown as green colored nodes). Interestingly our network places the anaerobic respiration regulator FNR in the cluster C 20, right at the top of the hierarchy. The main role of FNR is to regulate the switch from aerobic to anaerobic growth. FNR is also known as a global transcriptional regulator. The *fnr* gene regulates the transcription of a variety of functional genes, including chemotaxis, cell structures, temperature resistance, acid resistance, molecular biosynthesis and many more ([Lazazzera et al., 1993], [Spiro and Guest, 1990]).

FNR and two component systems are the two major regulatory systems that respond to a decrease in O_2 (oxygen) levels in *E. coli*. *fnr* has an O_2-sensitive $Fe-S$ cluster that directly senses O_2 and regulates site-specific DNA binding [Kiley and Beinert, 2003]. Oxygen plays an essential role in regulating the cellular processes in bacterial cells. The consumption of oxygen during aerobic respiration results in the conservation of cellular energy.

Color	**Description**
Green	Transcription Factor / Repression
Dark Green	Two component system
Red	Heat/cold shock
Dark Blue	Glutamic acid decarboxylase(GAD) system
Purple	Osmosis
Bottle green	Energy derivation
Yellow	DNA damage
Dark Pink	ABC transport

Table 5.4: Color index of gene regulatory networks shown in Figure 5.12 and corresponding descriptions are over-represented GO terms from the clustering analysis.

From literature Kang et al. [2005] it is clear that *fnr* controls expression of $100s$ of gene and their products in *E. coli*. Kang et al. [2005] describes an analysis of the wild type *E. coli* MG1655 strain to determine which genes are differentially expressed in response to O_2 and/or FNR. Their findings suggested that 465 genes were regulated by changes in environmental O_2 and/or FNR. Like *fnr*, other transcription factor genes involved in the network shown in Figure 5.12, are *pspF*, *relE*,

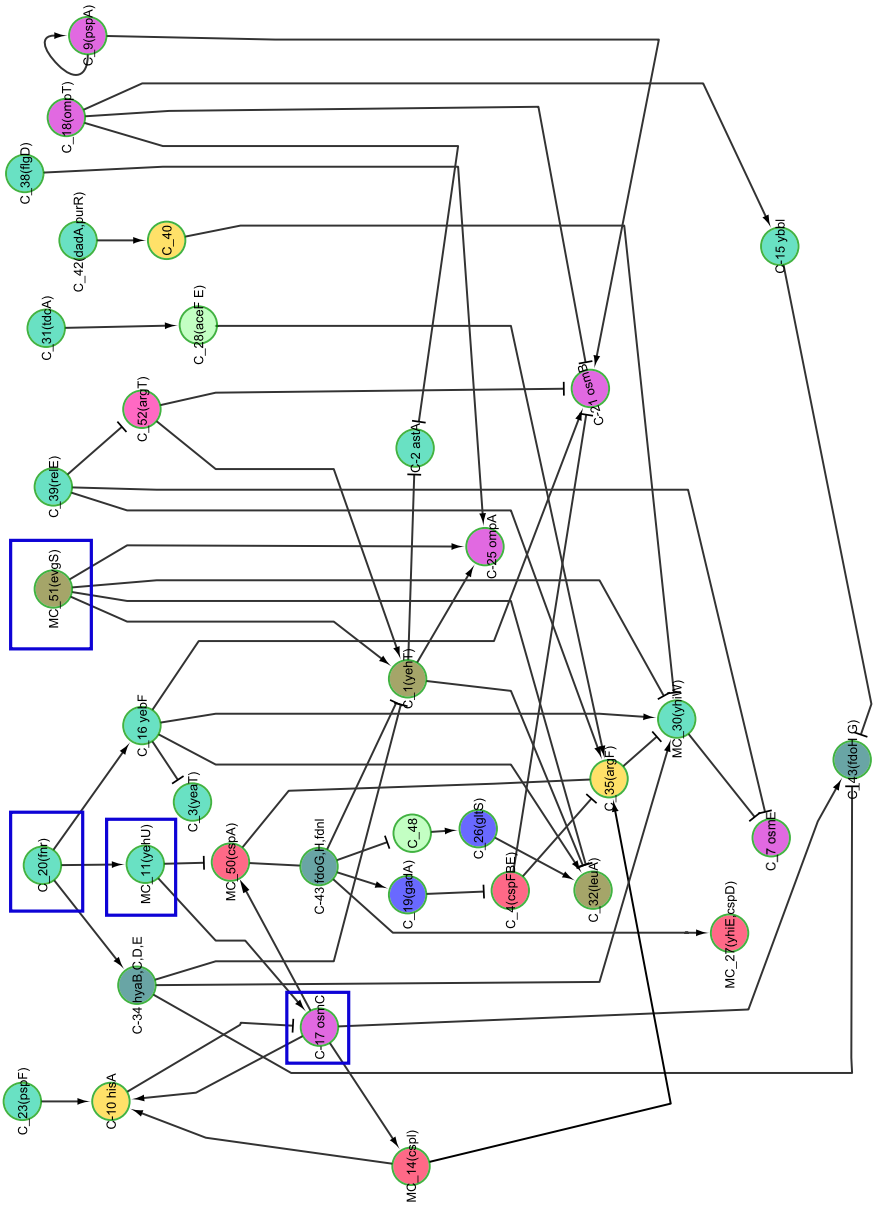


Figure 5.12: The inferred gene regulatory network using the temperature shift dataset. Here each node represents a cluster of genes and labels on the nodes show the cluster number with the representative gene. Arrows show positive interaction and $\perp$ represents negative interaction between two nodes. Colored frames around portions of network (green nodes on TF) are subject for detailed study in the following section.

tdcA, *dadA*, *purR*, *flgD*, *yehU* and *yeaT*. These genes can be identified in clusters shown as green color nodes of the network.

In the network shown in Figure 5.12 we can see that the cluster containing transcription factor *fnr* directly upstream of another cluster containing transcription factor *yehU*. *yehU* which inturn is negatively connected to a cluster containing the cold shock gene (i.e. *cspA*). It is very interesting to observe the down regulation of the cold shock inducible proteins is linked to energy regulator and its oxygen control. These are active in the temperature phase of $10 - 15°C$. The cold shock proteins *cspA,cspI* and *cspG* appear in the cluster C-14. Wang et al. [1999] describe *cspI* as the 9^{th} member of the cold shock protein family (also known as *cspA* family). Amongst all nine proteins from *cspA-cspI* the *cspA,cspB* and *cspG* are the cold shock induced proteins and *cspD* the stationary phase inducible protein. This represent an intersting hypothesis emerging from our model.

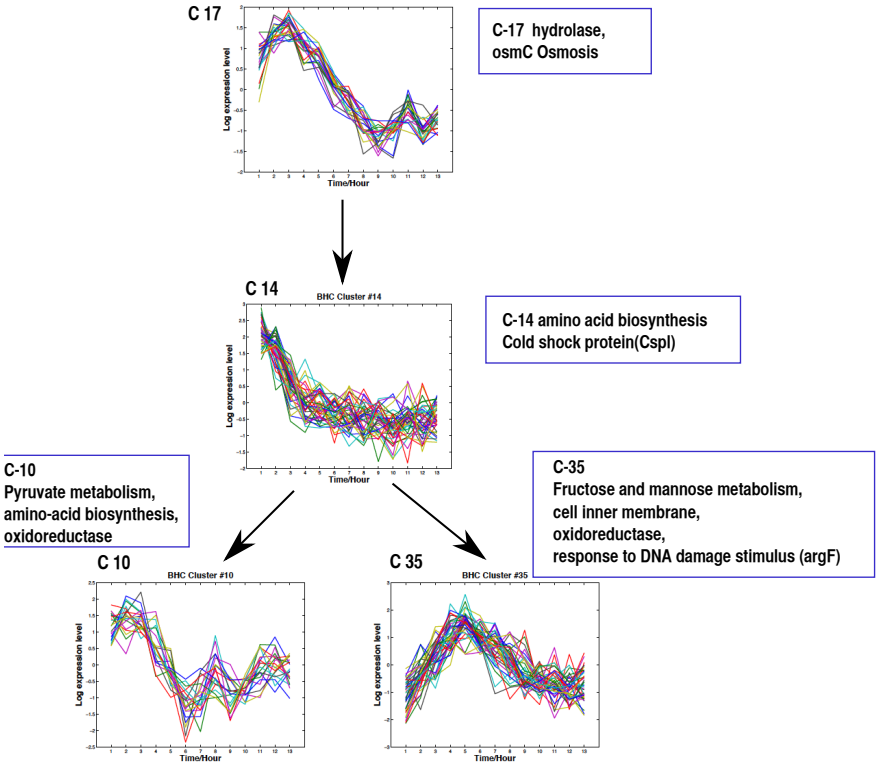


Figure 5.13: Sub-network downstream of cluster 17 from Figure 5.12

We isolate a few key subnetworks from the network in Figure 5.12, inorder to obtain a better understanding of the overall network. In Figure 5.13 shows another interesting sub-network, the cluster containing osmosensitive genes *osmC* and *osmY* is upstream of the heat/cold shock gene *Cspl*. *Cspl* is in turn upstream of the induction activation of the *hisA* and *argF* genes, which are known to be regulated in several environmental changes, including temperature difference. When *E. coli* culture is transfered from lower temperature to higher a set of cold/heat shock protein are induced [Jones et al., 1987]. These proteins are conventially classificed in two groups of based on their expression profile [Yamanaka, 1999]: class I proteins that includes family of cold shock proteins and class II includes DNA binding proteins, pyruvate dehydrogenase (i.e. pyruvate metabolism).

In *E. coli* the major outer membrane porin protein *ompA* functions to regulate osmotic pressure between the cell and its surroundings. A second sub-network is shown in Figure 5.14, this represent the TCS sensor *evgS* (cluster C-51). Our model predicts that *evgS* regulates the expression of the outer membrane protein OmpA [Kaeriyama et al., 2007], [Sato et al., 2000]. *evgS* is a putative regulator which encodes an integral membrane transporter of proline, glycine betaine and other osmoprotecting compounds [Eguchi et al., 2004].

A third sub-network is shown in Figure 5.15 we observe that *yehU* regulates the gene *osmC* which is a hub gene that further regulates other genes such as *hisA* that responds to any change in the environment. Enzymes such as formate dehydrogenase (*fodH* and *fodG*) appeared to be moderately affected by osmolarity [Gouesbet et al., 1993]. The expression of bacterial cold-shock proteins (CSPs) CspL and CspA are highly induced in response to cold shock, and some of them are essential for bacterial cells to gain growth at low temperature. This is also observed under other stress condition such as heat shock and osmotic stress [Stuebs et al., 2005]. It is observed that expression of CSPs such as CspL and CspA gets stimulates by osmotically inducible protein OsmC [Wang et al., 1999]. The case

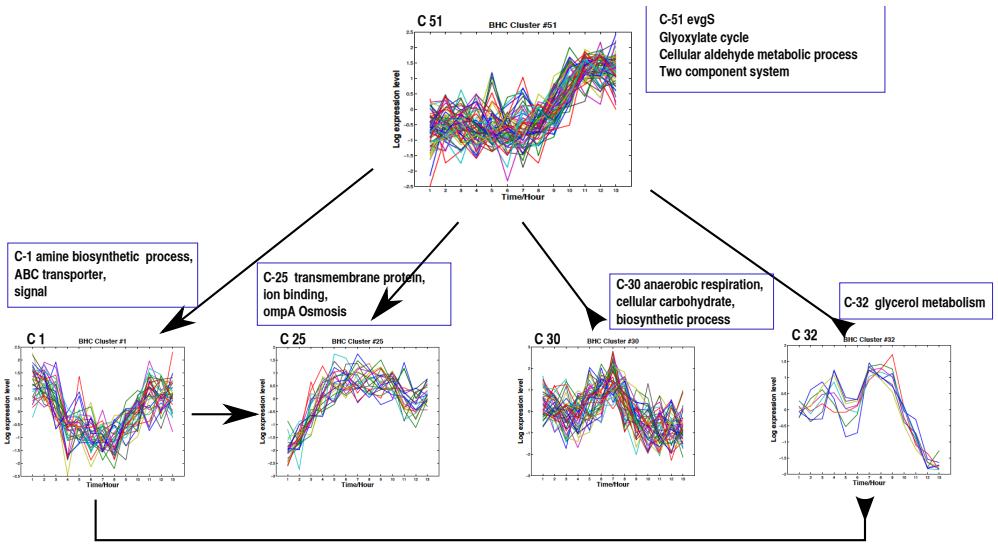


Figure 5.14: Sub-network downstream of cluster 51 from the regulatory network in Figure 5.12.

study by Weber et al. [2006] revealed the up-regulated protein PoxB under higher osmolality condition. It can be observed into early phase of adaptation by following the expression pattern of protein OsmC from cluster 17 and PoxB from cluster 10.

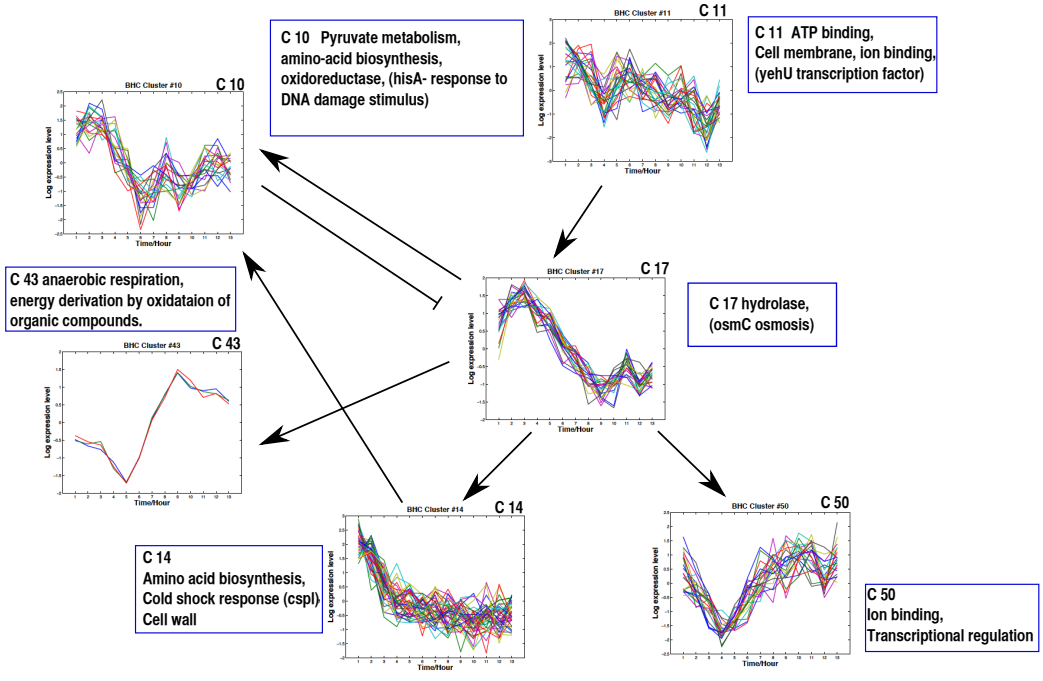


Figure 5.15: Sub-network downstream of cluster 11 from the network in Figure 5.12

With the help of literature Figure 5.16 highlights fourth subnetwork where *htpG* from C-1 acts as molecular chaperon that is activated in stressed *E. coli* cells. Thomas and Baneyx [2000] show that the absence of *cplB* (a gene from C-7) or *htpG* (from C-1) at $42°C$ leads to the increased aggregation of a fusion protein whose folding depends on *Dnak-DnaJ-GrpE*. Where *DnaJ* is a chaperon that controls the heat shock response in *E. coli*. Diamant and Goloubinoff [1998] described a temperature controlled activity of the *Dnak-DnaJ-GrpE* chaperon. In *E. coli* the *Dnak-DnaJ-GrpE* component chaperon system resolves denatured protein aggregations and assists the refolding of proteins via the *ATP/ADP* exchange factor. *DnaJ* (from C-51) binds the nascent protein that evolves from the ribosome, targeting *DnaK-ATP* to the protein. The complex of the polypeptide chain *DnaK-ADP-DnaJ* is stabilised by *ATP* hydrolysis. In this way it prevents the polypeptide from aggregation with other unfolded polypeptides. After *GrpE* resolved the complex by *ADP/ATP* exchange the polypeptide is released as an intermediate state of proteins mainly known as molten globule (MG). The conversion of a mature protein from the MG state requires a controlled folding process such as the *GroEL/ES* complex [Guisbert et al., 2004].

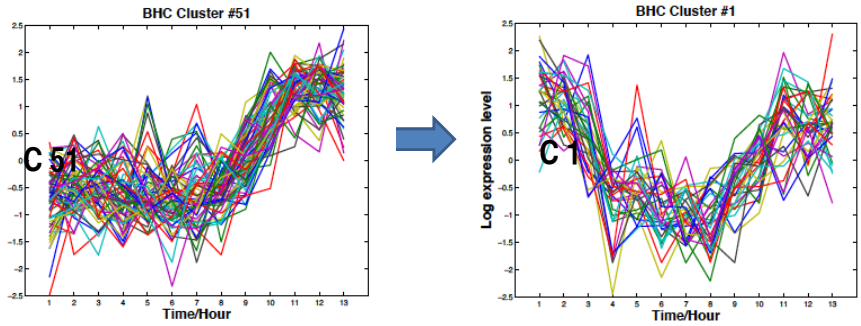


Figure 5.16: In this sub network we observe that the key regulator *htpG* acts as a molecular chaperon that is transcribed by *dnaJ* from cluster 51 in response to heat stress on *E. coli*.

There are smaller heat shock proteins such as *ibpA* and *ibpB* playing an important role to reduce the stress load of *E. coli*. Lethanh et al. [2005],Kuczyska et al. [2002] evaluate the impact of *ibpAB* deletion or over-expression on stress response. Deletion of the *ibpAB* operon can result in protein aggregation and in inactivation of enzymes (fructose-1,6-biphosphate aldolase) in cells under high temperature. The experiment proposed by Kuczyska et al. [2002] demonstrates that the *ibpAB* protein is essential for an extreme and long term heat shock response.

5.7 Summary

In this case study we have explored how data from temperature shift experiments can reveal an enormous amount of underlying information. The analysis of *E. coli* cells that undergo the temperature shift unfolds an adaptation process that shows a rapid transition between two equilibrium states. By using advanced Bayesian techniques we have reconstructed the structure of an hypothetical regulatory network and shown that relevent sub-network are biologically plausible.

The initial task of selecting differentially expressed genes from two condition datasets (i.e. control and temperature shift) was handled well by the GP2S method. However the top differentially expressed genes were thresholded and further investigated using a clustering approach. Clustering unveiled that many sets of genes regulated by the same operons remained intact in a cluster. However we have also observed that clustering smaller numbers of differentially expressed genes gives a more significant set of operons instead of a much larger set of differentially expressed genes. In the current case of larger sets of differentially expressed genes we have observed that single operon split between two or more clusters.

The inferred network highlights not only the heat or the cold sensitive genes but also genes those are involved in the osmosis might be responsible for adaptation to changes in temperature. Summarising four subnetworks suggests an experiment

to validatate. Further experiments may be required to confirm our hypothesis. For example the role of FNR controling cold shock response genes (e.g. *Cspl* gene) could be tested by micro-array analyis of an FNR mutant in control ($37°C$) and cold shock condition.

Chapter 6

Transcriptional and metabolic response of *E. coli* K-12 to acid adaptation

Pathogenic *E. coli* have to pass through the low pH environment in the stomach in order to mount an infection. Mechanisms that enable *E. coli* to survive in this low pH are thus potentially relevant for pathogenesis. So far three acid stress responses have been identified and studied in *E. coli* in pathogenic and non-pathogenic laboratory strains of *E. coli*. These three acid resistance (AR) systems are glucose repressed oxidation system (AR1), glutamic acid decarboxylase(GAD) system (AR2) and arginine decarboxylase system (AR3) [Castanie-Cornet et al., 1999],[Foster, 2004]. However the acid response system and their relationship between different pathways are unclear and are till poorly understood. Along with transcription data we have metabolic composition from the very same experiment that could provide a insight on system level to understand the important response mechanism of *E. coli*. With the help of our developed algorithm we study the transcriptional response and incorporate metabolite compositions on the basis their correlation with gene expres-

sion profile. For the correlation study between metabolites and gene expression we have made use of Gaussian process regression analysis.

The microarray data was handled from preprocess to postprocessing steps as described in the following sections. Section 6.1 introduces microarray profiling for the acid stress case study. In Section 6.1.1 we describe the microarray data pre-processing which was done by Anna Stincone and Dr Francesco Falciani at the University of Birmingham. In Section 6.1.2, the identification of differentially expressed genes is described. Section 6.3 describes clustering and eigengene analysis following a similar approach to that presented in Chapter (5) 5 Section 5.4. Section 6.6 is divided into two subsections; subsection 6.6.1 gives details about the computational experiment performed using the Gibbs sampler algorithm and Subsection 6.6.2 presents the inferred gene regulatory network and its interpretation in terms of AR systems. The latter also includes a detailed explanation of the plausibility of the inferred network. The summary of entire chapter is given in the Section 6.7.

6.1 Expression profiling by microarray.

This section describes the experimental setup for the *E. coli* K-12 (strain MG1655) response during adaptation from pH= 7 to pH= 5.5. The design of this experiment is shown in Figure 6.1. The experimental details are given by [Stincone et al., 2011] and are summarised here. The bacterial strains were cultured in lysogeny broth(LB) media. In this experiment the initial pH of the sample culture was maintained neutral (i.e., at pH = 7). Later on, in order to generate an acid shock the pH was decreased to pH = 5.5 by adding *hydrocloric acid* (HCl). In order to recover cell pellets this sample underwent icing and then was centrifuged for 10 minutes. Stabilized cells were then recovered and stored at $-80°C$. The RNA was isolated using the Quiagen Rneasy® kit. Labelling of RNA was done with Cy5 labelled dCTP by using the CyScribe Post-Labelling kit. After purification

mRNA was hybridized overnight and then washed in AdvaWash and scanned with ScanArray GX using the Scan Array software. To study the dynamics of acid adaptation in *E. coli* at pH= 5.5 as opposed to that at pH= 7, samples were collected from a continous culture system for transcriptomics. Two pilot samples were collected from the continous culture system i.e., at pH= 7 for 30 min and 1 hour. Adaptation to acid shock was then monitored at 5min intervals for 13 time points (T = $30sec$, 5, 10, 15, .., 60 $mins$). This experiment was repeated for three biological replicates. Under this experiment single channel hybridization was carried out using $Cy5$ dye and total 4217 gene expression measurements over 15 time points including the initial control samples.

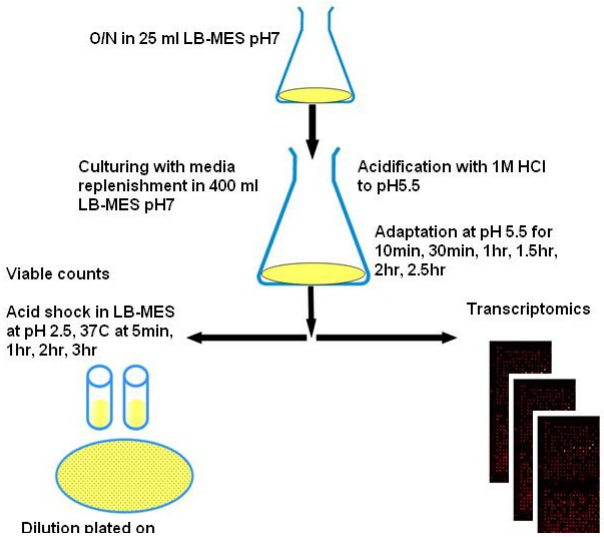


Figure 6.1: The experimental system for acid stress condition.

6.1.1 Quantile normalization

To correct for systematic errors in the single channel array data, they were normalized using quantile normalization (Irizarry et al. [2003]). The idea behind this method comes from a quantile-quantile (Q-Q) plot. When the Q-Q plot shows a straight diagonal line it is concluded that the two data vectors follow the same

distribution. Otherwise if the Q-Q plot is not a straight line then the data vectors do not follow identical distributions. This concept was then extended for an $n-$dimensional dataset in a way that all n data vectors will follow an identical distribution if by plotting the quantiles in n dimensions, a straight line along the unit vector $(1/\sqrt{n}, \cdots, 1/\sqrt{n})$ line is observed. This indicates that a set of data could follow the same distribution by projecting the points of the $n-$dimensional quantile plot onto the diagonal.

Therefore quantile normalization assumes that the intensities of each chip originate from the same underlying distribution. This implies that the quantile for each chip is the same. However, biases in the signal generating process result in chip-specific distributions. The goal of quantile normalization is to remove these biases by transforming the data such that each quantile is the same across all chips. The acid stress data was normalized using quantile normalization from preprocessCore R package.

6.1.2 Identifying differentially expressed genes

6.1.2.1 Using the timecourse Package

Tai and Speed [2006] present a multivariate empirical Bayes method that provides analysis of differential expression for microarray time course data including replicates. The authors derive a multivariate empirical Bayes statistic (the MB-statistic) for one and two samples in order to rank genes. The ranking of differentially expressed genes takes into account replicate information from the time series experiment. This is carried out by testing a null hypothesis,

$$\begin{aligned} H_0 : \boldsymbol{\mu} &= \boldsymbol{\mu}_0 I, \ \boldsymbol{\Sigma} > 0, \qquad \textit{gene expression level constant} \\ H_1 : \boldsymbol{\mu} &\neq \boldsymbol{\mu}_0 I, \ \boldsymbol{\Sigma} > 0 \end{aligned} \tag{6.1}$$

where $\boldsymbol{\mu}_0$ is the expected value of the expression of the gene and I is a vector of $1's$.

To test the hypothesis a standard likelihood ratio (LR) statistic was used. Implementation of different statistics across genes reduces the number of false positives and false negatives. Therefore together with $MB-$statistic the authors also present the $\tilde{T}^2$ statistic and Hotelling T^2 statistic. This method is implemented using the statistical programming language R as the *timecourse* package and includes functions for identifying differentially expressed genes from replicated microarray time course experiments with one or more biological conditions. Figure 6.2 represents a typical example of the *cadA* gene which ranked 1^{st} using timecourse analysis. By using timecourse analysis we gathered approximately 2000 differentially expressed genes out of a total of 4217 genes.

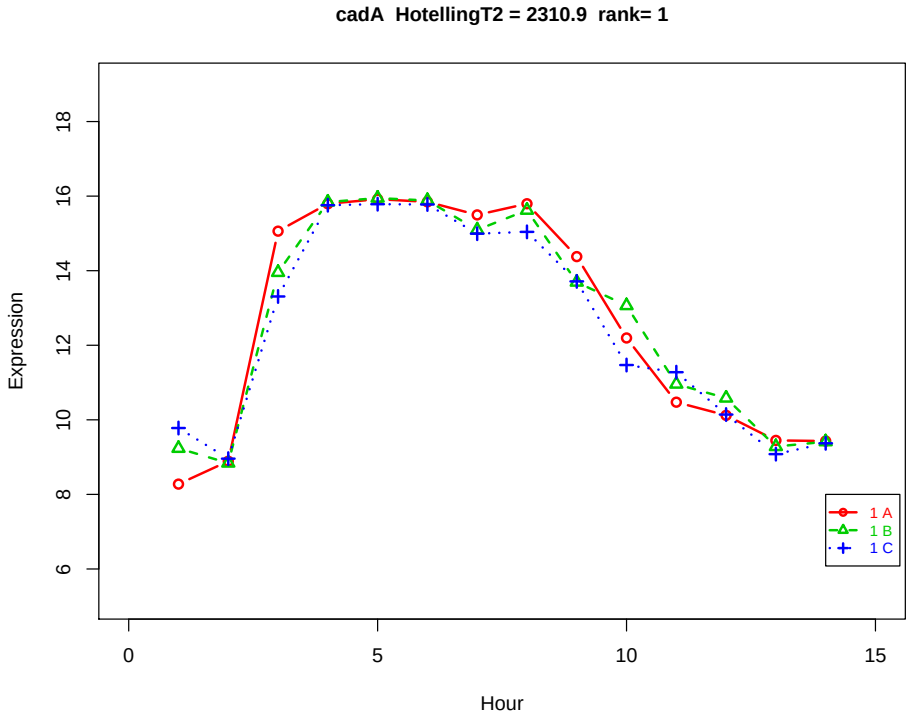


Figure 6.2: An example result produced from timecourse analysis using the acid stress dataset. Three curves labelled as "A", "B", "C" represent three replicates. The title of this figure gives more detail about the gene name, its corresponding Hotelling T^2 score and ranking. This figure represents the expression of the *cadA* gene whose Hotelling score is 2310.9 and it is ranked first out of total 4217 expression measurements.

So far we considered the transcriptional response of *E. coli* cells under acid stress. In the following section we introduce and include the metabolite measurements that were obtained from the *E. coli* acid stress experiment. Our goal here is to combine the gene expression measurements with the metabolite measurements to study the cell response under the acid stress.

Metabolomics is the qualitative and quantitative study of all low molecular weight compounds present in the cell. They are required for the maintenance of cell growth and for other normal cellular functions. The metabolome consist of small organic molecules such as amino acid, fatty acids, carbohydrates, vitamins, and lipids.

We begin with an introduction to the Nuclear Magnetic Resonance (*NMR*) technique, which was used for the metabolite profiling described in Section 6.2. Collective metabolite profiles were then explored using the BHC clustering algorithm described in section 6.3.1. Section 6.5 describes the use of Gaussian process regression (GPR) method for correlation analysis between transcriptomic and metabolomoic responses. Combining gene expression measurements with the metabolite concentrations we have attempted to reconstruct a combined transcriptional-metabolite network. Our objective here is to address the role of metabolites in the response to *E. coli* under the acid stress.

6.2 NMR profiling

The one dimensional and proton decoupled *NMR* spectra were converted to an appropriate format for multivariate analysis using MATLAB. Each spectrum was then segmented into 1600 chemical shift bins between 0.2 and 10.0ppm, corresponding to bin widths of 0.005 ppm (i.e., $2.5Hz$). The spectral area within each bin was then integrated to get $1 \times N$ vector containing intensity based descriptors of the original spectrum. The study suggests, the chemical shift bin between $4.7 - 5$ ppm contains

residual water peak (K), which was removed. The total of $N - K$ remaining bins were normalised and log transformed after adding a constant to address smaller values [Viant, 2003].

Viant et al. [2003, 2005] describe the application of *NMR* spectroscopy to assess to the biological stresses on an organism from the environment and to visualize toxic action during the embryogenes of fish. Results from these experiments confirm that *NMR* based metabolomics can distinguish the biochemical profiles of different sample groups. Viant [2003] proposed an improved method for the reading and interpretation of *NMR* metabolite data. The improved method was achieved by the simplification of two dimensional $2D$ J-resolved spectroscopy (JRES) NMR spectra which contain less resonance for the same number of metabolites. The JRES spectrum projects the chemical shift and spin-spin coupling on different axes, in such a way that chemical shift axes cover less protein- decoupled $1D$ ^1H *NMR* spectra.

6.3 Data exploration

In this section the results of the clustering analysis using the BHC algorithm and the calculation of eigengenes (EG) using singular value decomposition (SVD) technique are presented. The derivation and implementation of both methods were discussed in detail in Chapter (5) 5 Section 5.4.2 5.4 (EG section).

6.3.1 Clustering transcriptional profiles

The top 1000 differentially expressed genes were clustered using the BHC software [Cooke et al., 2011]. For this analysis we used the BHC clustering algorithm with a covariance functioned based on both cubic splines and the squared exponential function. The clustering algorithm was set to run in the presence and absence of outliers. Table 6.1 shows the total number of clusters obtained using the two proposed covariance functions in the presence and absence of outlier measurements.

Outlier method	Squared exponential	Cubic spline
Absence	67	70
Presence	65	69

Table 6.1: Number of clusters obtained from the BHC algorithm set up with two different covariance functions in the presence and absence of outlier measurements.

The smallest number of clusters (i.e. 65) was obtained using a squared exponential covariance function and by incorporating the outlier measurements. The heatmap output is shown in Figure 6.3. The dendrogram on the left (in blue lines) represents the merging of clusters and red dotted lines represent clusters that were rejected before merging by the BHC algorithm. The red dots show where tree is split.

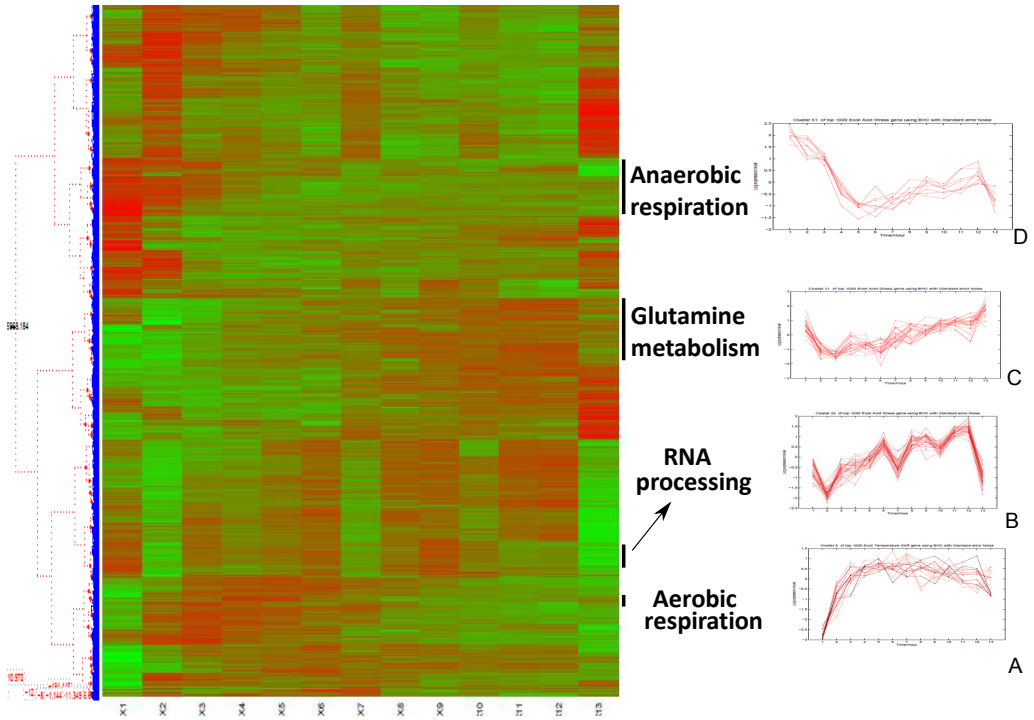


Figure 6.3: Heatmap representation of BHC clustering output. The dendrogram representation of the cluster output is shown on the left of the heatmap. The red dotted lines over the dendogram show the merges rejected by the algorithm. On the right shows the biological processes shared with the indicated clustered gene profiles are shown.

The clustering results from BHC combine the transcriptional events over the

period of adaptation and summarise it into 65 clusters of gene expression profiles. Four clusters of interest are highlighted on the right side of Figure 6.3. The two clusters A and D represent the early response to stimulation including up- and down-regulated genes respectively. In these two clusters the major change has been observed in the first five minutes after introducing the acid. Moreover, the two other clusters B and C represent genes that are gradually up- or down-regulated in response to the acid stress.

6.3.2 Clustering with metabolite profiles

For the current study with NMR spectroscopy the metabolite intensities were measured by using chemometric (calibration) technique. Prior to any further data analysis the NMR spectrum were binned and most of peaks were collected from NMR data. This experiment was performed at Dr. Francesco Falciani′s and Dr. Mark Viant′s laboratories of the University of Birmingham. There are a total of 58 sets of metabolite profiles from the acid stress experiment. However only 10 out of 58 are identified metabolites.

Interestingly, in response to osmotic shock the concentration of amino acid metabolites is expected to decrease however, the concentration of *valine, leucine* and *isoleucine* increases after acid shock as in metabolite cluster 2 in Figure 6.4. A rapid decrease in the concentration of *glycine betaine* was observed when exposed to the low pH as in metabolite cluster 5 in Figure 6.4. Moreover glutamate was expected to increase in concentration as shown in the metabolite cluster 6 of Figure 6.4. These initial observations draw our attention to a few key players which have an important role in the the adaptation mechanism under low pH.

Eigen gene analysis

The eigenmetabolites (EM) were calculated using singular value decomposition as described in Chapter 5 5. Figure 6.5 represents the principal eigenmetabolite profiles corresponding to the 10 clusters resulting from BHC (as in Figure 6.4).

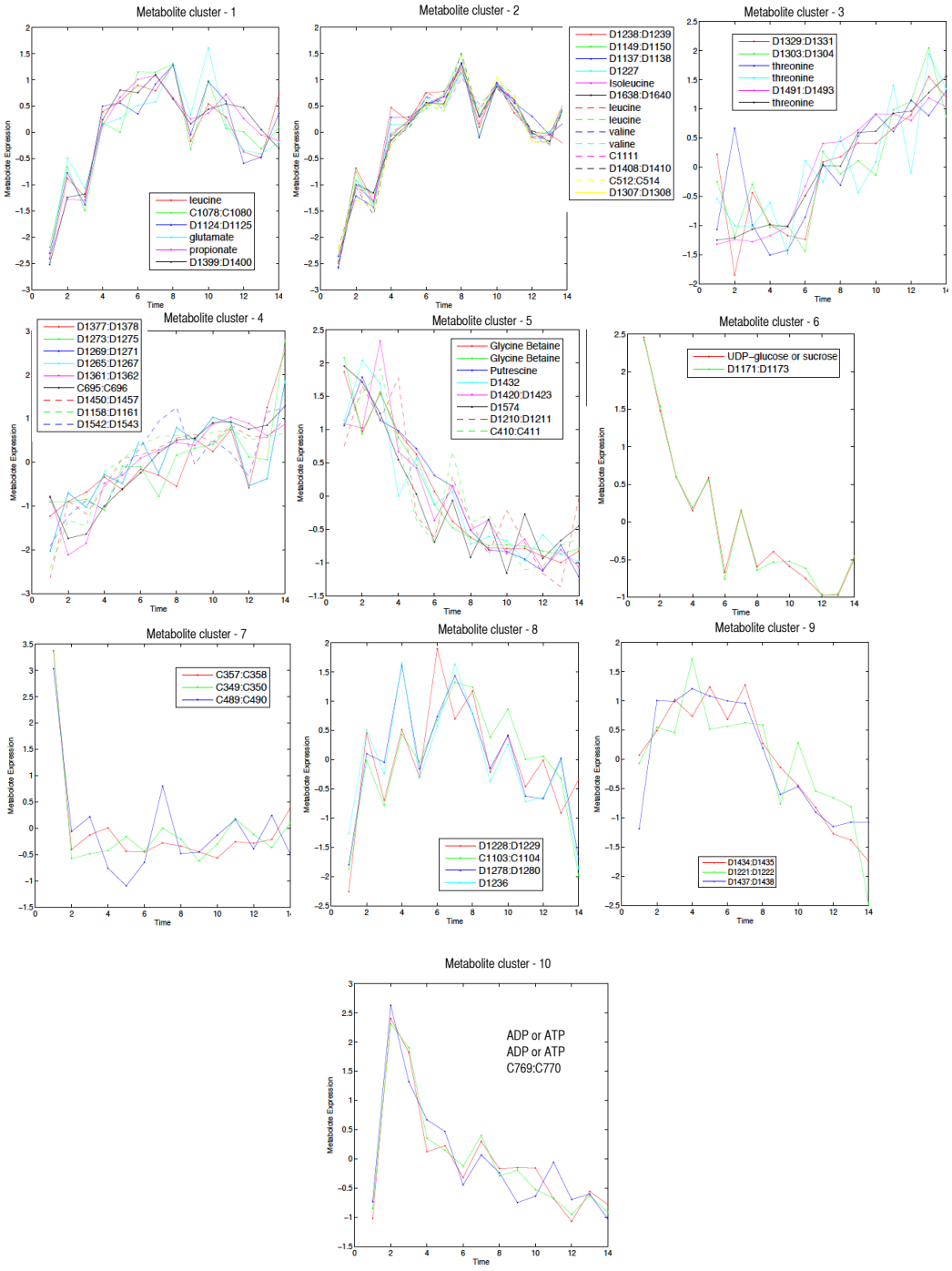


Figure 6.4: Metabolite clusters with the identity of the metabolites involved in each cluster reported in the legend on the right of each plot.

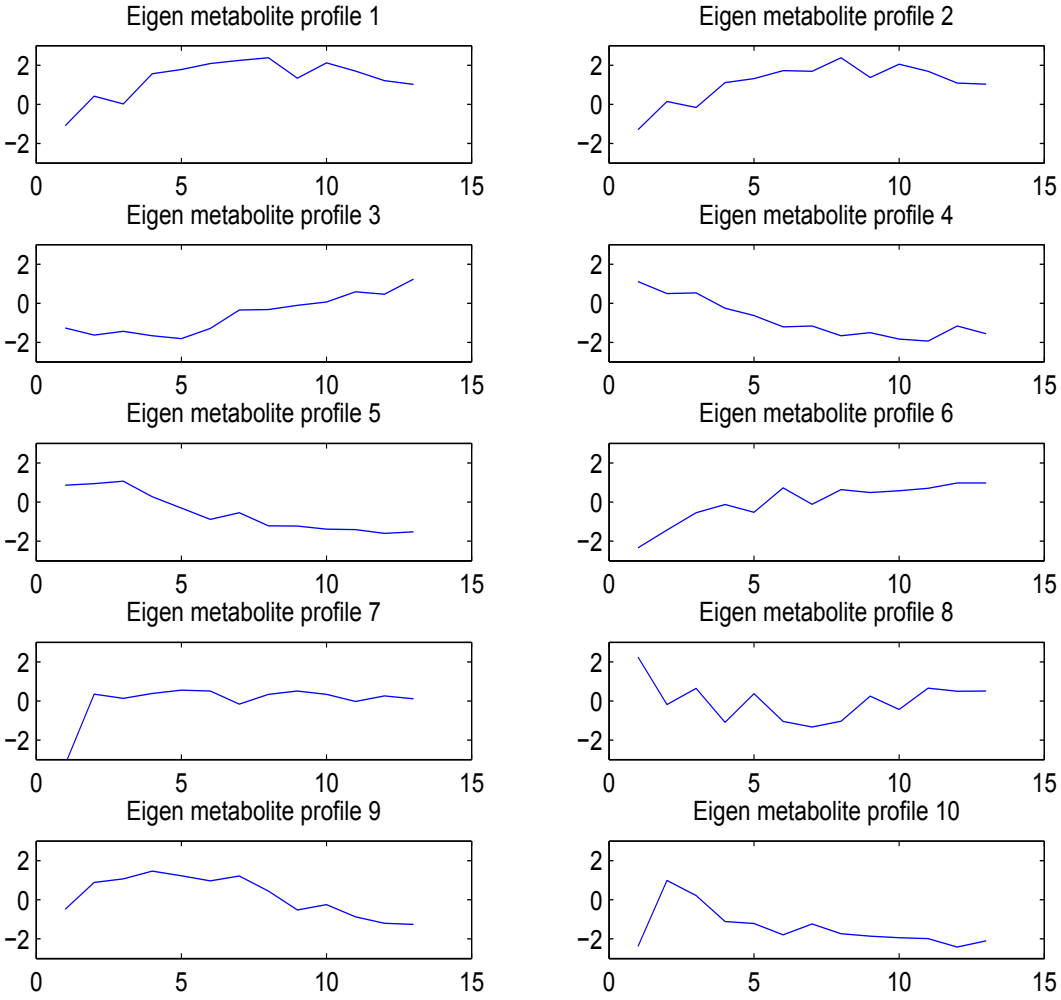
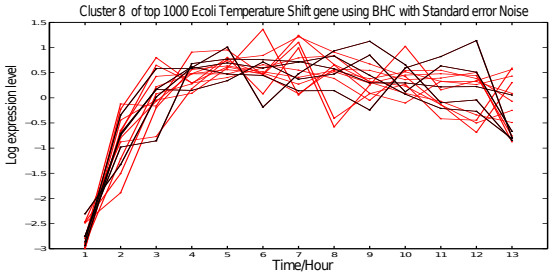


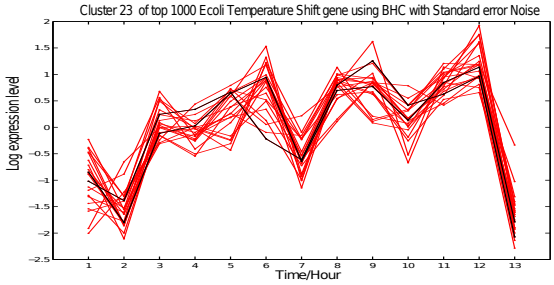
Figure 6.5: Eigen metabolite profiles of the 10 clusters resulting from the BHC algorithm.

6.4 Functional annotation clustering

In order to test whether clusters of correlated genes represent a coordinate functional response we used functional enrichment analysis, of the gene ontology (GO) as described in Chapter 5 5. In this section we discuss a few interesting annotations from the overall annotations of the 65 clusters. Figure 6.6 shows the example annotation of the 8^{th} and the 23^{rd} cluster.



Cluster #	Annotation Cluster	Enrichment score	representative term	p-value	FDR
8	1	3.95	aerobic respiration	7.5E-9	4.0E-7
8	2	3.76	tricarboxylic cycle	1.1E-3	8.5E-3



Cluster #	Annotation Cluster	Enrichment score	representative term	p-value	FDR
23	1	2.13	anerobic respiration	1.7E-5	1.8E-3
23	2	1.74	nucleotide binding	3.7E-5	2.8E-3
23	3	1.3	peptidoglycan-based cell wall	3.4E-3	4.4E-2

Figure 6.6: Graphical representation of the functional annotation of the 8^{th} and 23^{rd} cluster. The top figure represents the gene expression profiles included in cluster 8. The table below the plot profile represents the functional annotation with the descriptional of genes involved in molecular/biological/chemical process. Among the list of up-regulated genes from cluster 8 we found enzymes *spy*, *cyoC*, *mqo* involved in aerobic respiration and are highlighted in black profiles. From the list of genes from cluster 23 we gather the enzymes *hyfH*, *nuoM* involved in anaerobic respiration and are highlighted in black colored profiles.

After GO enrichment analysis we followed the functional annotation clustering method to concatenate the biological terms on the basis of significant biological function. Funtional annotation clustering method is described in detail in Chapter 5 Section 5.4.1. The following tables 6.2 summarise the results from the annotation clustering for each of the 65 clusters.

Table 6.2: Annotation summary resulting from functional annotation clustering method. Gene annotation for each cluster are sub-divided into three most significant clusters with the term of highest significance shown. Column named "Clust ID" is cluster id, "Anno clust" is annotation clustering, "GO-BP" are gene ontology biological processes, "INT" are INTERPRO based annotation, "KEGG" are Kyoto Encyclopedia of Genes and Genomes, "representative terms" specifies the functional behaviour, "p-value" (≤ 0.05) and Benjamini's false discovery rate (≤ 0.01).

Clust ID	Anno Clust.	ES		Representative Term	P˙value	FDR
1	1	5.6	GP˙BP	di-, tri-valent inorganic cation transport	1.0E-08	5.1E-07
1	2	1.75	KEY	ion transport	1.4E-03	1.5E-02
1	3	1.65	KEY	transmembrane protein	5.8E-05	1.3E-03
5	1	1.48	KEY	cell inner membrane	1.9E-04	7.5E-03
6	1	12.64	GP˙BP	tricarboxylic acid cycle	1.9E-16	3.1E-15
6	2	4.67	KEGG	Citrate cycle (TCA cycle)	2.9E-08	7.5E-07
6	3	2.52	INTE	ATP-citrate lyase succinyl-CoA ligase	3.6E-06	9.7E-05
7	1	0.96	KEY	transmembrane protein	1.6E-03	4.3E-02
8	1	3.95	GP˙BP	aerobic respiration	7.5E-09	4.0E-07
	2	3.76	GP˙BP	tricarboxylic acid cycle	1.1E-03	8.5E-03
10	1	3.04	KEY	amino-acid biosynthesis	1.6E-04	1.2E-02
	2	2.08	GP˙BP	energy derivation by oxidation of organic compounds	2.9E-03	3.4E-02
12	1	2.13	KEY	signal	4.1E-05	3.0E-03
17	1	2.8	KEY	transport	8.6E-04	6.2E-03
17	2	1.92	KEY	cell membrane	2.1E-04	6.2E-03
18	1	6.53	KEY	cell inner membrane	2.4E-12	1.2E-10
18	2	3.66	KEGG	ABC transporters	9.9E-07	9.5E-05
18	3	1.83	GP˙BP	ion transport	6.3E-04	7.1E-02
19	1	3.43	GP˙BP	ciliary or flagellar motility	1.3E-05	1.2E-03
19	2	2.45	KEY	cell cycle	1.1E-03	2.3E-02
19	3	1.06	GO˙CC	organelle inner membrane	2.0E-03	1.4E-02

Continued on next page

Table 6.2 – *Continued from previous page*

Clust ID	Anno Clust.	ES		Representative Term	P˙value	FDR
20	1	1.21	GO˙CC	cell wall	1.4E-03	7.4E-03
21	1	1.46	KEY	membrane	2.1E-05	8.3E-04
23	1	2.13	GP˙BP	anaerobic respiration	1.7E-05	1.8E-03
23	2	1.74	KEY	nucleotide binding	3.7E-05	2.8E-03
23	3	1.3	GO˙CC	peptidoglycan -based cell wall	3.4E-03	4.4E-02
25	1	1.46	KEY	cell membrane	2.1E-04	6.0E-03
30	1	3.21	GP˙BP	anaerobic respiration	9.9E-06	7.4E-05
	2	3.15	GP˙BP	cytochrome complex assembly	1.6E-09	9.6E-08
	3	2.96	KEGG	Fructose and mannose metabolism	3.1E-04	2.3E-02
31	3	1.58	KEY	cell membrane	7.3E-06	4.4E-04
34	1	4.12	KEY	cell membrane	6.6E-10	4.5E-08
35	1	1.29	KEY	cell inner membrane	1.9E-04	6.3E-03
37	1	3.25	GP˙BP	nitrate assimilation	1.6E-06	7.0E-05
	2	3.11	GP˙BP	tRNA processing	9.2E-03	7.7E-02
	3	2.4	KEY	cell membrane	4.3E-06	9.9E-05
43	1	2.07	KEY	cell inner membrane	6.5E-05	1.6E-03
44	1	3.97	GP˙BP	glycerol metabolic process	1.0E-09	3.1E-08
	2	2.32	GP˙BP	carbohydrate catabolic process	2.4E-03	4.3E-02
45	1	6.92	GP˙MF	ion binding	3.2E-07	8.0E-06
45	2	3.29	GP˙BP	anaerobic respiration	7.3E-04	2.4E-02
45	3	3.13	KEY	oxidoreductase	1.8E-08	4.5E-07
46	1	1.53	KEY	transcription regulation	1.0E-02	8.1E-02
48	1	2.39	KEY	sugar transport	9.6E-07	6.4E-05
48	2	2.15	GP˙BP	carbohydrate catabolic process	1.9E-07	2.0E-05
	3	2.03	KEY	cell membrane	2.2E-05	7.4E-04
49	1	3.45	KEY	transport	1.1E-08	5.5E-07
	2	1.63	KEY	cell membrane	3.2E-05	3.3E-04
51	1	5.17	GP˙BP	anaerobic respiration	2.7E-07	1.4E-06
51	2	3.99	GP˙MF	iron ion binding	3.2E-07	8.3E-06
	3	3.69	KEY	oxidoreductase	4.8E-11	1.1E-09

Continued on next page

Table 6.2 – *Continued from previous page*

Clust ID	Anno Clust.	ES		Representative Term	P˙value	FDR
52	1	4.8	GP˙BP	anaerobic respiration	6.0E-10	2.4E-08
52	2	3.59	GP˙BP	energy derivation by oxidation of organic compounds	4.1E-09	8.5E-08
	3	3.47	GP˙BP	fermentation	2.2E-08	3.1E-07
53						
54	1	2.54	KEY	flagellum	4.2E-04	1.6E-02
55						
56	1	4.28	KEY	Histidine biosynthesis	1.4E-07	5.7E-06
58	1	2.73	KEY	cell membrane	7.0E-06	3.0E-04
58	2	2.68	KEY	nucleotide-binding	2.1E-05	3.7E-04
58	3	1.88	KEY	two-component regulatory system	1.1E-04	1.6E-03
59-60						
61	1	1.84	KEY	carbohydrate metabolism	1.6E-03	7.4E-02
61	2	1.48	KEY	amino-acid biosynthesis	3.1E-03	4.9E-02
63	2	1.65	KEY	pyridoxal phosphate	3.5E-03	7.3E-02
64	1	2.86	GP˙BP	glucose metabolic process	5.8E-04	2.2E-02
64	2	2.42	KEGG	Glycine, serine and threonine metabolism	9.9E-04	4.5E-02
65						

6.5 Gaussian Processes Regression Analysis

Before proceeding with the integration of mRNA and metabolite time series, this section describes the study of a correlation analysis between mRNA and metabolite profiles using Gaussian process regression (GPR) analysis. For the GPR analysis we have used the *gpml* toolbox provided by Rasmussen and Williams [2006]. A Gausian process is completely specified by a mean function and a positive definite covariance function (also known as Gaussian or linear kernel). It is possible to embed Automatic Relevance Determination (ARD)[1] directly into the covariance function

[1]The GP hyperparameters such as the correlation length-scale can be determined by using maximum likelihood technique.

in following way;

$$K_{i,j} = cov[f(x_i), f(x_j)] = K(x_i, x_j)$$
$$K(x_i, x_j) = exp(\frac{1}{2}\sum_{\zeta=1}^{d} k_\zeta (x_i^\zeta - x_j^\zeta)^2) \tag{6.2}$$

where $k_\zeta > 0$ is the ARD parameter. By estimating $k_\zeta > 0$ we determine the relevance of i^{th} feature of the EMs in the prediction of the j^{th} feature of the target EG, and x_i^ζ denotes the ζ-th element of x_i. Lower the ARD parameter value the higher this feature correlates with the target.

Use of the GP toolbox

The GP toolbox contains a single user function called "gp", with additional support structures and functions. On the basis of the user′s requirement these function can be addapted and used for data analysis. For example the inference method provided in this tool is a function that computes the approximate posterior, the approximate negative log evidence and its derivatives with respect to the hyperparameters given a specific model and a data set. For our analysis we have specified the likelihood function to be Gaussian and made use of the exact inference method.

For demonstration purposes initially we made use of 27 EGs, these EGs resulting from BHC clustering of the top 200 differentially expressed genes from the acid stress experiment. Each targeted EM was then regressed against the EGs. Given randomly defined initial values of the hyperparmaters of the model, the algorithm was set to iterate for 5000 iterations and repeated 3 times with randomly chosen initial starting points. Figure 6.7 provides typical results from the GP regression analysis.

In a similar fashion we regressed all 10 EMs, one by one against 65 EGs, these EGs resulting from the BHC clustering of the top 1000 differentially expressed genes. The results from the Gaussian process regression analysis is sumarised in the following Table 6.3.

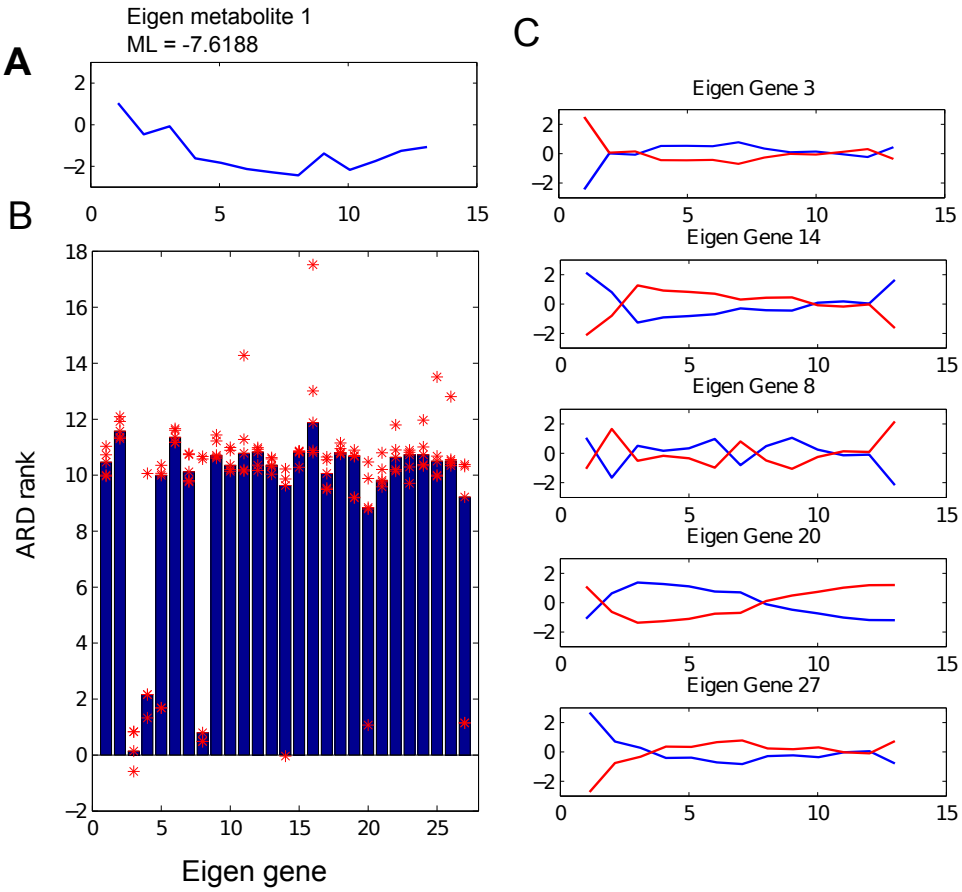


Figure 6.7: Panel A: represens the target eigen metabolite and evaluated model evidence on top. Panel B: Red dot represents three samples of ARD parameter and blue bars are the median of the ARD parameter. Panel C: shows top 5 lowest ranked ARD index and corresponding EG profiles. Blue profile is an actual EG and red profile is an inverse of blue (because GP is non-linear process that allows inversion and rotation).

Here we give some biological validation that supports the correlation analysis between EGs and EMs. Eigen metabolite 1 contains glutamate which through glutamate dehydrogenase synthesize α-ketoglutarate which is an intermediate in the tricarboxylic acid cycle EG 8 (6.8). Glycine from EM 5 (conserved glycines [Szentpetery et al., 2004]) and nitrate assimilation (this includes the uptake and transport into cells by nitrate transporters) from EG 37 are part of ATP-binding cassette(ABC) transporter proteins. EM 6 indicating composition of UDP-glucose is

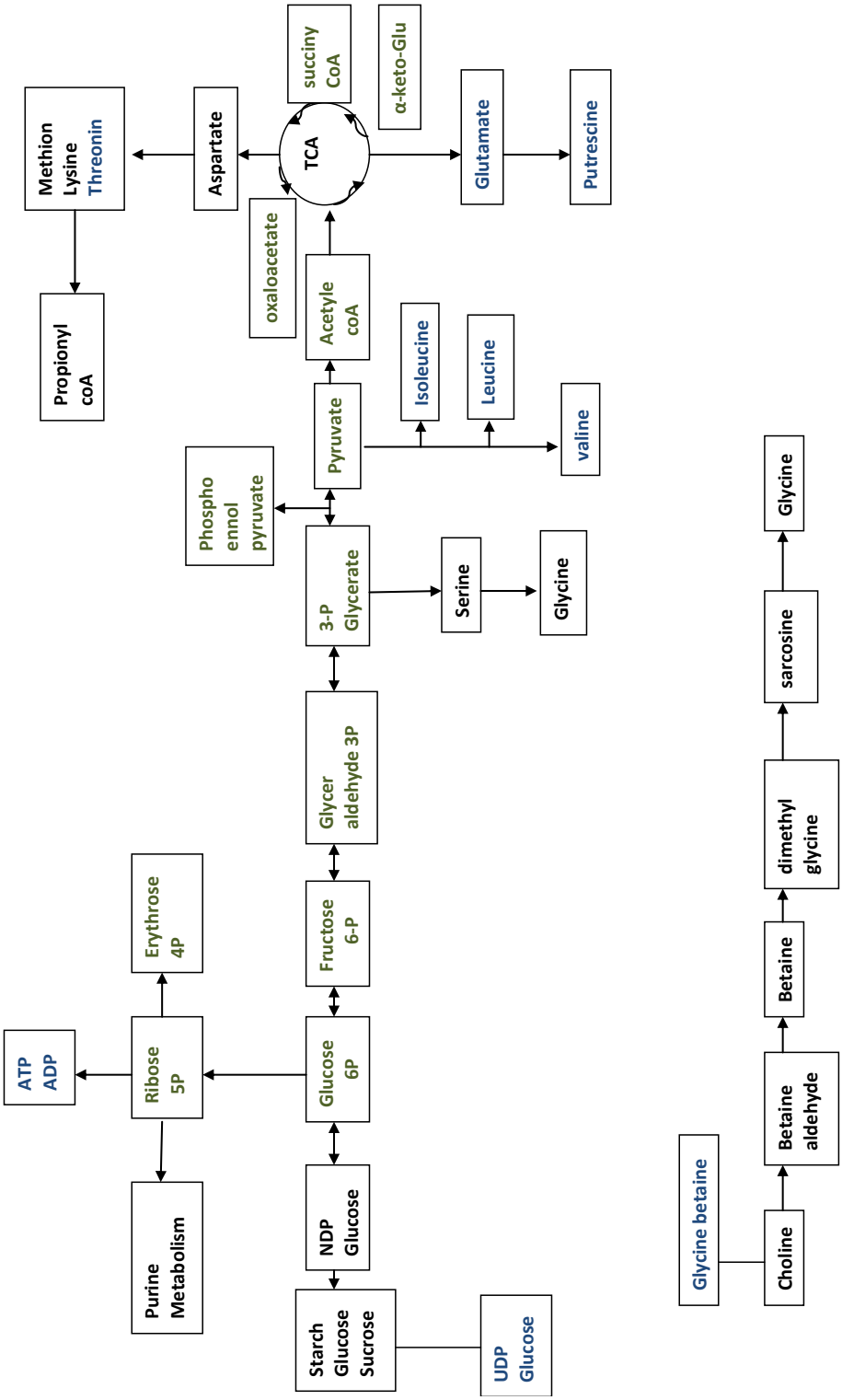


Figure 6.8: A simplified version of the *E. coli* metabolic map representing the identified metabolites in this study Neidhardt et al. [1990, chapter 5]. Identified metabolites from acid stress experiments are labeled in blue text can be located on this map.

Eigen Metabolites	Eigen Genes	Marginal likelihood
1 (leucine,glutamate,propionate)	8 (aerobic respiration, tricarboxylic acid cycle)	-6.2214
2 (Isoleucine,leucine,valine)	62 (transition metal ion binding,	-6.0920
	cell inner membrane)	
3(threonine)	40	0.9886
4(unknown protein)	40	-4.7074
5(glycine betaine, putrescine)	37 (nitrate assimilation,	-1.9613
	tRNA processing)	
6(UDP-glucose)	52 (anaerobic respiration,	-6.5602
	fermentation)	
7(unknown protein)	48 (sugar transport,	-5.9369
	carbohydrate catabolic process)	
8(unknown protein)	38 (cell inner membrane)	-14.2060
9(unknown protein)	5 (cell inner membrane)	1.4088
10(ADP or ATP)	40	2.0022

Table 6.3: The correlation between EMs and EGs including identified metabotites and fuctional annotation based in the analysis described in Section 6.4. The last column of marginal likelihood indicates that the lower the value more reliable the resulting correlation between EMs and EGs.

a precursors of fermentation which represented in the EG-52. The most comnonly used series of reactions for oxidizing glucose, also known as Embden-Meyerhoff-Parnas pathway (EMP). Thus we have observed that the Gaussian processes regression analysis provides a yardstick for correlation analysis to bridge metabolic data with the transcriptional.

6.6 Inference of regulatory network

6.6.1 Numerical experiment

After studying the functional annotation in detail we proceed with the network inference following the approach described in Chapter 5 Section 5.4.1. For the network inference the dataset consists of 65 EGs over 13 time points for three biological replicates. The MCMC algorithm developed in this thesis was set to run for increasing

hidden state space dimensions (i.e., $k = 1, 2, \cdots, 10$). Each model was then set to run for at least 5 independent Markov chains for iterations $> 150,000$.

The convergence was then studied by visual inspection and by measuring the PSRF (as described in Section 2.3.1) values for each chains of the model. Panel **A**, **B**, **C** of Figure 6.9 shows the PSRF value calculated for dynamic parameters **A**, **D** and for noise parameter **R**. It is observed that the PSRF measurements lies within or below the convergence acceptance margin (i.e., a straight line at 1.1). Different colors in the plots represent increasing hidden state space dimension.

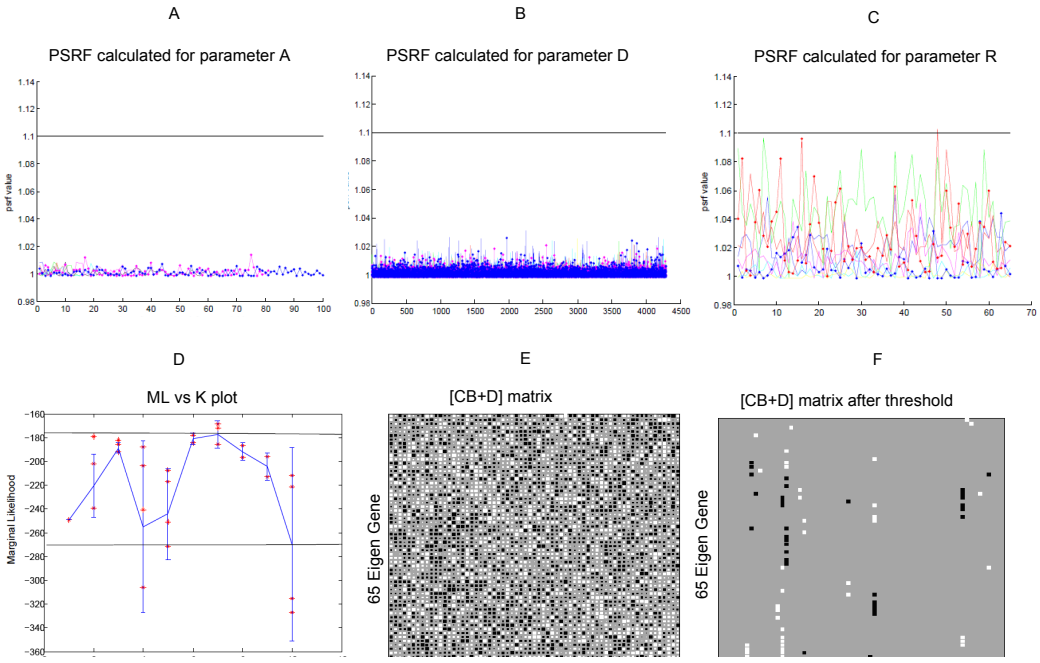


Figure 6.9: Summary plot results from MCMC output. Panel A,B,C shows the *psrf* calculated for parameter **A**, **D** and **R** respectively. Different colors represent model parameters from increasing dimension of the hidden state. Panel D represents model evidence versus hidden state. Panel E shows the Hinton diagram of the $[\mathbf{CB} + \mathbf{D}]$ matrix. Panel F shows the Hinton diagram after thresholding.

Once convergence was confirmed the model evidence was calculated using the MCMC output according to Chibb's method (as described in Section 2.4). The plot in panel D represents the marginal likelihood against increasing hidden state space dimension. The model evidence was then used to decide on the optimal value

of the hidden state space dimension that best fits the data. From Figure 6.9 panel D, the optimum hidden state space dimension (i.e., giving the maximum value of the marginal likelihood) appears at $k = 7$. Using the estimated parameters of the model with $k = 7$, the $[\mathbf{CB} + \mathbf{D}]$ matrix was calculated. This represents the gene-gene interaction matrix of dimension 65×65. Panel E shows the Hinton diagram of the estimated $[\mathbf{CB} + \mathbf{D}]$ matrix. The network was pruned by using the Z-statistic test with a suitable significance value. Figure 6.9 panel F shows the Hinton diagram obtained with a Z-score test with confidence level of value of 95%.

6.6.2 Results and discussion of inferred network

The eigengene network of *E. coli* shown in Figure 6.10, reveals several interesting structures and dynamical features of gene expression during acid stress. With the help of the annotation study from Section 6.4 and reviewing in the literature we present the interpretation of the EG network.

Color	Description
Green	Transcription Factor / Repression
Dark Green	Two component system
Red	Heat/cold shock
Purple	Osmosis
Bottle green	Energy derivation
Yellow	DNA damage
Pink	ABC transport
Dark gray	Amino acid metabolites
Dark blue	Metabolites

Table 6.4: The color index representing the biological process of nodes of inferred eigengene network shown in the Figure 6.10.

Preliminary study of the overall inferred network structure is based on potentially interesting biological processes. Moreover for more detailed analysis we zoom into the part of network to make it more informative. Each node of the network in Figure 6.10 represents a cluster of genes. Based on the functional annotations we have highlighted the nodes of the network and reported them in table 6.4. For

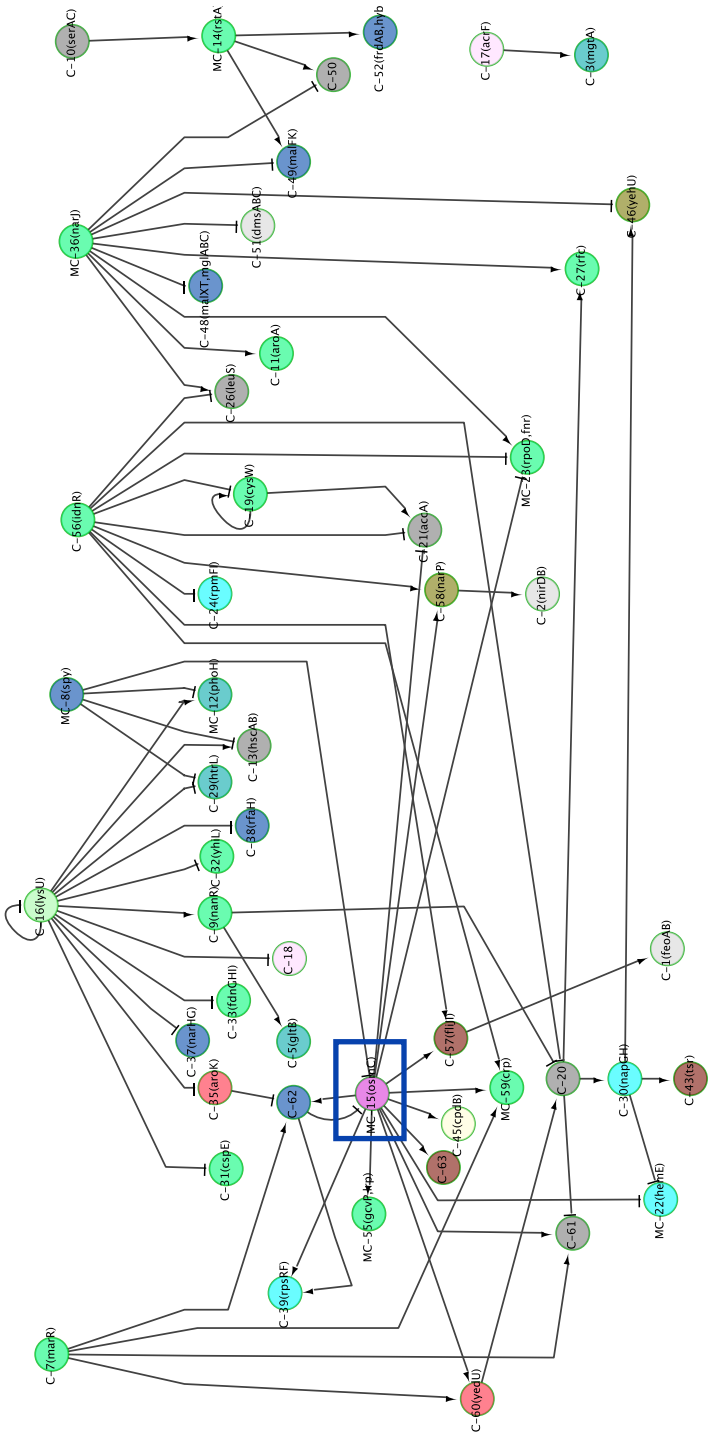


Figure 6.10: Inferred gene regulatory network using acid stress dataset. Here nodes represents the clusters with highlighted significant genes. The arrows and $\perp$ sign shows positive and negative interaction between two nodes respectively. Colored frame around portions of network is subject for detailed study.

example the transcriptional activity is labelled in a green color. Transcriptional activitiy indicates very first step of gene expression, in which a fragment of DNA copied into RNA with the help of RNA polymerase enzyme. Two component systems highlighted in dark green nodes are the basic stimulus-response mechanism that sense and responds to the environmental changes. The heat/cold shock proteins are highlighted in red these are class of functional proteins those are involved in the folding and unfolding of other proteins. Hence their expression indicates the exposure of cell to the extreme temerature including other stress. Similarly other biological functions are given in the table 6.4.

In order to interpret the network we will discuss different aspects of the model following a top to bottom strategy. On the basis of our current biological understanding of environmental stress response we have observed first the signal (e.g. acid exposure) was sensed by a two component system, then the signal was passed to global regulator and then to effector systems such as AR systems. We have observed the osmotic induction of *osmC* gene expression in node 15 of the network that could be interesting to study the adaptation mechanism of *E. coli* under acidic stress.

Two component system (TCS)

Basically TCS involves stimulus-response coupling mechanism that enable bacteria to sense, respond, and adapt to changes in their environment or intracellular state. In *E. coli* almost 50 different TCSs are present, some of them are mentioned in table 6.5. The sensor kinase *ArcB* and response regulator *ArcA* of an textitArc system found co-expressed in cluster C-61. The response regulator of nitrate respiration (*NarP*)and Pho regulation (*PhoB*) appeared in cluster C-58 and C-12 respectively. The porin regulation through environment signal of osmotic pressure is yet another interesting observation and worth going indetail.

For the osmotic pressure control the relative levels of proteins are *ompC* and *ompF* and are found in node C-15 in Figure 6.10. These proteins are found in the

outer membrane of *E. coli* *OmpC* and *OmpF* are porin protein that allows metabolites to cross the outer membrane of gram-negative bacteria. If osmotic pressure is low then the synthesis of *OmpF*, a porin with a larger pore, increases; if osmotic presser is higher then the *OmpC*, a porin with a smaller pore is made in larger amounts. The response regulator of this systems is *OmpR*. When *OmpR* is phosphorylated it activates the *OmpC* gene and represses transcriptional activity of *ompF* gene [Kaeriyama et al., 2007].

System	Environment signal	Sensor Kinase	Response regulator	Activity of response regulator
Arc system	Oxygen	ArcB	ArcA	Repressor /activator
Nitrate respiration	Nitrate and nitrite	NarX	NarP	Activator /Repressor
Pho regulation	Inorganic Phosphate	PhoR	PhoB	Activator
Porin regulation	Osmotic pressure	EnvZ	OmpR	Activator /Repressor

Table 6.5: Example of two component systems that regulates transcription in *E. coli*(adapted from [Madigan et al., 2009, Chapter 9]).

Global transcription factor

An organism often needs to regulate many unrelated genes simultaneously in response to a change in its environment. Regulatory mechanisms that respond to environmental signals by regulating the expression of many different genes are called the global control system. Both the lactose operon and the maltose regulon respond to global controls in addition to their own specific regulation. It is interesting to identify global transcription factors those are differentially expressed such as *fnr* in node 23, *fis, lrp* (leucine-responsive regulatory protein) in cluster 34 and node 55 respectively, *crp* (catabolite repression protein) in node 59, in acid stress experiment [Lazazzera et al., 1993].

There are four acid stress response systems (ARs) that are known till date,

enables *E. coli* to survive under a low pH. Three out of four ARs depends on the extral supply of glutamate, arginine and lysine amino acids. These three ARs shares same basic mechanism of reductive decarboxylation of the amino acid by a consumption of a proton, followed by extrusion of the product from cytoplasm through antiporter that also imports the original amino acid. These three ARs are explained in detail below:

AR1: Glucose decarboxylase

AR1 is based on FoF1-ATPase and are active in the absence of amino acids [Richard and Foster, 2003] [Stincone et al., 2011]. The expression of the glucose repressed ARs requires the alternative sigma factor *rpoS*. The *rpo* operon appears in clusters 23 and 24 [Bhagwat et al., 2006]. In addition to the glucose repression AR1 system also depends on global transcription factor *crp*. The Catabolite repression is a global control system that helps cells makes the most efficient use of available carbon sources. It is also known as glucose "effect" because glucose was the first substance shown to initiate this response. Some organisms require carbon from other sources than glucose and this causes catabolite repression. The key point is that the substrate that represses the use of other substrate is a better carbon and energy source. In this way, catabolite repression ensures that the organism uses the best available carbon and energy source first. Despite its name in catabolic repression transcription is controlled by an activator protein and is actually a form of positive control. The activator protein is called the cyclic AMP receptor protein (CRP). A gene that encodes a catabolite-repressible enzymes is expressed only if CRP protein binds to DNA in the promoter region. This then allows RNA polymerase to bind to the promoter [Small et al., 1994].

AR2: glutamate decarboxylase

In AR2 glutamate is the substrate , *gadA*, found in the 15-kb region is known as the

Acid Fitness Island (AFI) and is requires for resistance at low pH. One of *gadA* or *gadB* genes encodes glutamate decarboxylase and *gadC* that encodes the putative glutamate. It is observed that the *gadB* is up-regulated under low pH. In the GRN inference *gadB* appears in cluster 15 along with the presence of the *osmC* gene [Castanie-Cornet et al., 1999] [Opdyke et al., 2004].

The node−15 from the inferred network carries an interesting group of genes that has shown resistance to acid stress. The up-regulation of *gadB* from cluster 15 also indicates the acid resistance activity. Glutamate decarboxylase(GAD) is one of three acid resistance systems that we have mentioned earlier. Following reviews by Foster [2004] on *E. coli* acid resistance, the decarboxylase dependent AR systems seemingly have a simple mechanism of action. The GAD enzymes *GadA* and *GadB* (from AR2) contains phosphate enzymes that replaces the carboxyl group of their amino acid substrates with a proton that is transported from the cytoplasm. In this process CO_2 is produces along with γ-amino butyric acid (GABA) as shown in the Figure 6.11-a.

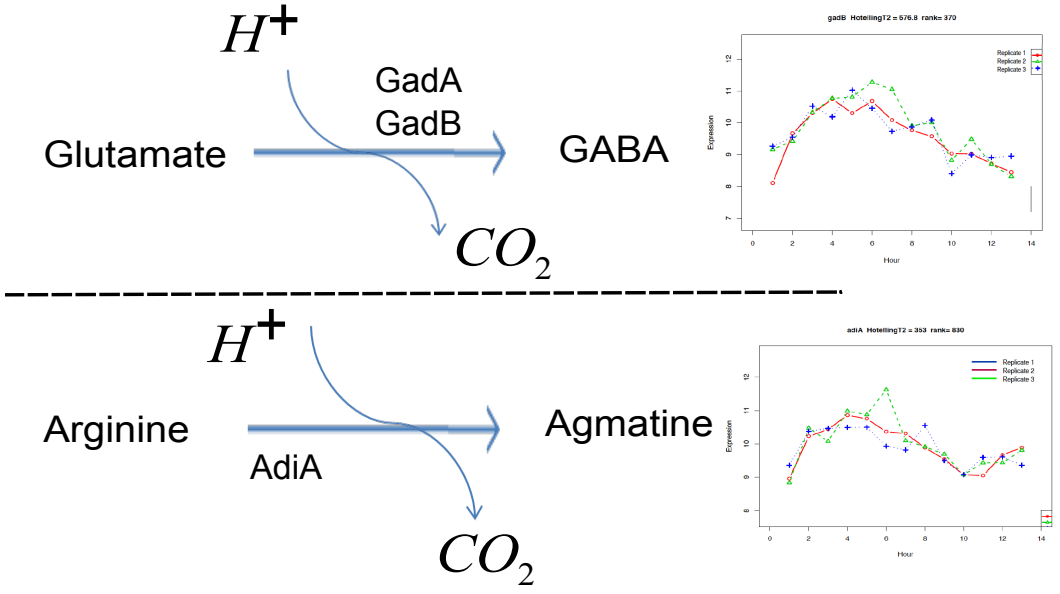


Figure 6.11: Consumption of proton($H+$) during decarboxylation of glutamate and *arginine*.

AR3: Argine decarboxylase

The second AR system requires expression of the structural gene for arginine decarboxylise decarboxylise *adiA*. The up-regulation of *arginine* appears in cluster 14. In figure 6.10 cluster 14 indicates the presence of arginine decarboxylise in the regulatory mechanism of *E. coli* under low pH [Richard and Foster, 2004]. The *AdiA* from C-14 arginine with a proton that is transported from the cytoplasm produces CO_2 leaving agmatime as an end product, shown in Figure 6.11-b.

Osmotic Pressure

The osmotic pressure in the bacteria cell allows a measure of the concentration of small free solute molecules contained within the semipermeable plasma membrane. This concentration provides a certain type of internal environment in which any necessary chemistry of the cell takes place. Moreover the difference between the osmotic pressure from within the cell and that of the medium (i.e., the differential osmotic pressure) determines the firmness of the plasma membrane, i.e. how firmly the membrane is pressed against the rigid peptidolycan layer. Thus such differential osmotic pressures may affect one or more of the several membrane-related activities.

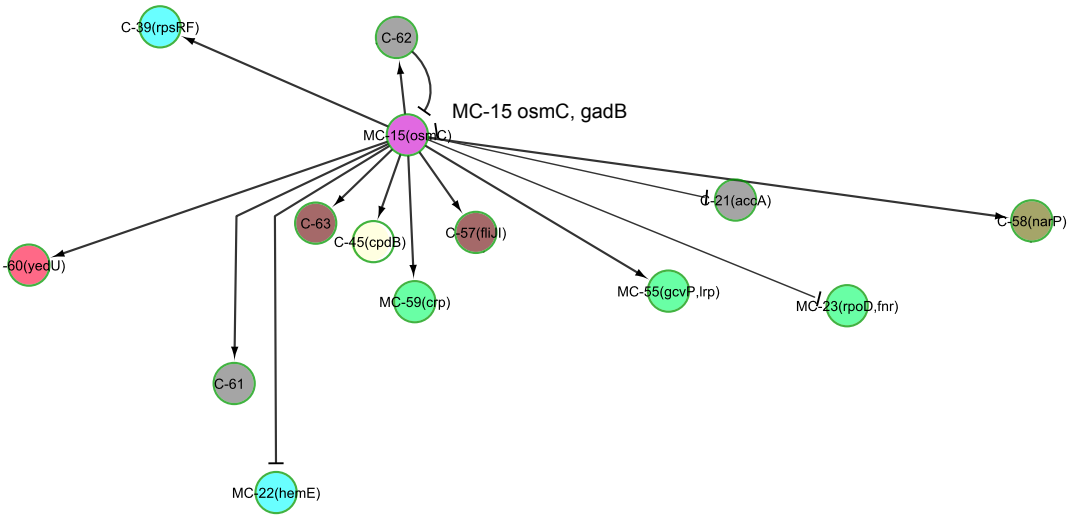


Figure 6.12: Sub network from GRN 6.10.

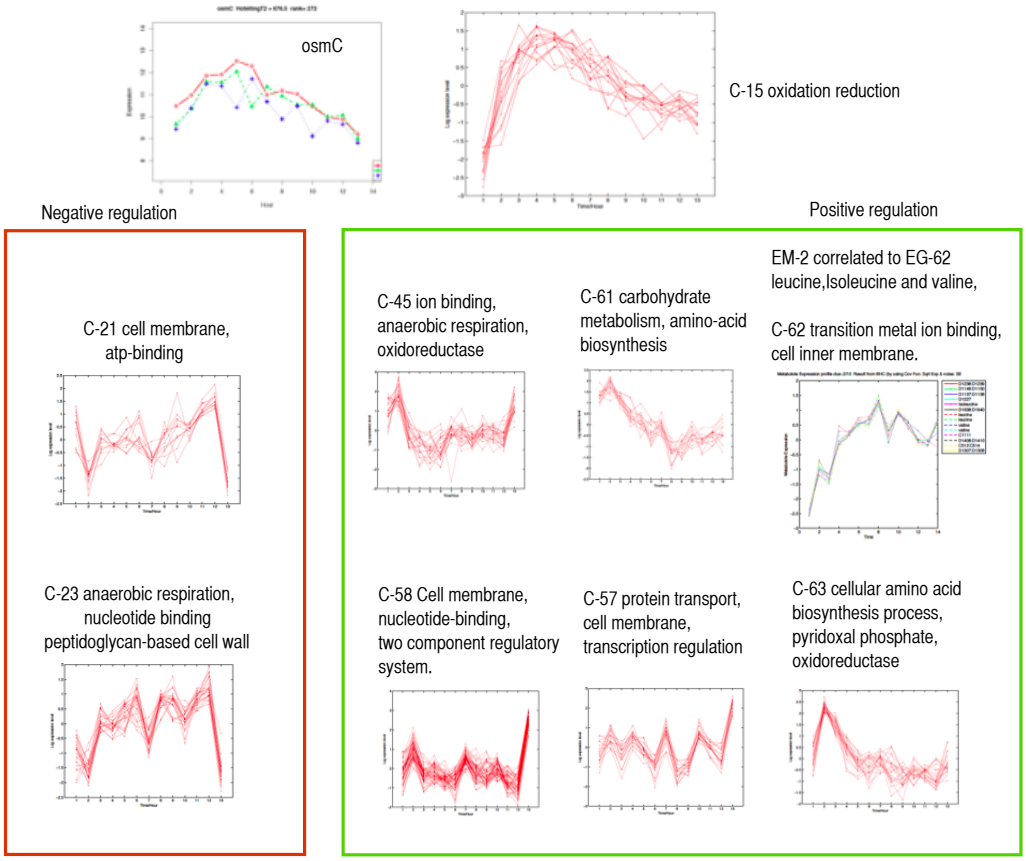


Figure 6.13: Alternative to figure 6.12 with expression profiles of clusters included.

The expression of *osmC* is growth phase-regulated and osmotically inducible Gunasekera et al. [2008]. The sutdy on *osmC* mutants was carried out by [Wang et al., 2007] and have shown increased sensitivity to oxidative stress. In figure 6.12 our model predictes the influence of *osmC* on other biological processes such as anaerobic respiration, amino-acid biosynthesis(like leucine, Isoleucine, valine), cell membrane protein, etc., for more detail see figure 6.13. Figure 6.13 is an alternative representation of Figure 6.12 where we group positive and negative regulation of *osmC* in green and red box respectively. It is observed that under acid stress the activation of an oxidoreductase enzyme (as a catalyst) regulates the transfer of electrons from one molecule to another [Record et al., 1998].

6.7 Summary

In this case study we have shown the application of a reverse engineering approach to understand the response and adaptation mechanism of *E. coli* under acid stress. We begin our analysis with the pre-processing of gene expression measurements by using timecourse analysis. Through this analysis we were able to threshold differentially expressed genes from the entire dataset. Selected differentially expressed genes were then clustered using the BHC algorithm. Clustering gives quick overview of the entire biological experiment. Studying functionality of co-expressed genes allowed us to represent each cluster by its eigen gene representative. By inferring the EG network we obtained an understanding of the response mechanism of *E. coli* at system level.

Similar to the transcriptome data clustering analysis, metabolite compositions were also clustered using BHC clustering algorithm. Eigen metabolites were defined as a representative of clustered metabolites. For the integration of EMs with EGs we have used Gaussian process regression analysis. The correlation between EMs and EGs were justified by looking into their functional annotations.

The inferred network using our developed algorithm shows the presence of acid resistance system that have been studied so far. In addition our model predicts that many genes involved in the response of osmotic pressure showed immediate response under acidic conditions. This could be experimentally validated and some links between acid and osmotic stress could be worth establishing. However we believe, the overall aspect of the adaptation mechanism of *E. coli* k-12 to low pH is in its infancy and requires further analysis.

Chapter 7

Conclusion

In the first part of the thesis we present a computational framework based on Bayesian MCMC methods (GBSSM) to address parameter learning, inference and model section tasks. This approach allows exploring the unknown parameters of the model from full Bayesian conditional distributions. The robustness of inference can be examined by various validations based on data generated from simulated data and *in silico* networks.

In the second part of this thesis we present applications of the developed GBSSM algorithm to high-throughput post-genomic data sets. The results from this approach provide biologically plausible hypotheses that can be experimentally validated. In the following Section 7.1 we summarise the contributions of this thesis and in Section 7.2 we discuss possible future work.

7.1 Summary of Contributions

Chapter 2 provides the theoretical background for the development of a Gibbs sampling algorithm for SSMs with and without feedback. This chapter demonstrates the derivation of the posterior distributions by using conjugate priors, in order to infer the parameters and the hidden variables of the model. The extended Gibbs sampling

algorithm incorporates learning of hyperparameters by integrating Metropolis Hastings steps within Gibbs sampling. The replicates of gene expression measurements are explicitly incorporated in the algorithm. However the task of defining a suitable hidden state dimension is crucial and difficult. This might affect the accuracy of the infered gene-gene interactions. In order to address these issues we make use of the model evidence as a yardstick for model selection. The model evidence that we calculated uses the Gibbs sampling output following Chib's method [Chib, 1995].

In **Chapter 3** we demonstrate the validation of the proposed GBSSM algorithm. Initially a numerical experiment was performed using simulated data generated from a toy model network. We compare the GBSSM algorithm with constant hyperparameters for the noise against that with inferred hyperparameters. This experiment shows that hyperparameter learning assists the Gibbs sampler to achieve convergence towards a stationary distribution.

Chapter 4 begins with a review of an *in silico* network that represents plausible biological processes. The numerical experiment was performed by using the developed algorithm to reverse engineer this *in silico* network. This chapter also demonstrates how the model selection to determine the optimal dimension of the hidden state space was done on the basis of marginal likelihood. At the end of the chapter we compare GBSSM with a variational Bayesian approach (VBSSM) using a receiver operating characteristic analysis. The following chapters address some real-world problems of of inferring GRNs from microarray data using the GBSSM algorithm.

Chapter 5 presents a first application to microarray gene expression data. In this case study we investigate the response and adaptation mechanism of *E. coli* bacterial cells under temperature shift. Results from this case study not only provide a useful application of the GBSSM algorithm but also unveil many aspects of the

regulation of *E. coli* undergoing a shift to higher temperature. Some of the regulatory aspects of this process have been well studied and experimentally verified. On the basis of the structure of the underlying regulatory networks, for both the adaptation and response, we propose the following hypotheses, which can be used for future biological knock-out experiments.

Hypothesis 1: The application of network inference analysis using GBSSM reveals the potentially interesting hypothesis that there is a connection between osmotic stress responses and those of temperature shift. Experimental validation could be performed by the analysis of knockout mutants of the osmoprotectant enzymes, such as *osmC*. The literature on *E. coli* suggests that *osmC* is regulated by *osmR*. Therefore along with mutating *osmC* it would be interesting to mutate *osmR*. Significant change (or reduction) in the growth of *E. coli* bacteria under temperature shift after knocking out *osmC* and *osmR* would support this hypothesis.

Hypothesis 2 : In addition, the application of the GBSSM network inference technique leads us to hypothesise that *cplB* and *htpG* play a key role in the adaptation to cold and warmth respectively. This can be experimentally validated by performing knockout experiments based on mutating the *cplB* and *htpG* genes. If the hypothesis holds this will make *E. coli* intolerant to heat or cold shocks.

Chapter 6 presents a case study that investigates the adaptation and response of *E. coli* cells under acid stress. The objective of this case study is to represent acid stress adaptation in non-pathogenic *E. coli* strains. There are three acid resistance systems (ARs) studied so far. We observe the presence of these ARs as a part of the inferred network structure.

Recent work of Stincone et al. [2011] describes a study of the *E. coli* K-

12 strain BW25113 under acid stress. This study hypotheses that OmpR (i.e. a response regulator for osmoregulation) may be regulating the complex transcription involved in acid adaptation and this hypothesis was validated by performing a mutant OmpR experiment. In addition the authors describe the possibility of some relevance of the osmotic stress response to the adaptation mechanism under low pH. Our inferred network from GBSSM also highlights the role of the stress-inducible membrane protein OsmC regulating the transcriptional activity involved in the adaptation mechanism under acid stress. This indicates potentially interesting knockout experiments based on mutating *osmC* gene in the *E. coli* K-12 MG1655 strain under acidic conditions.

In Chapter 6 we also describe metabolite profiling of *E. coli* cells under acid stress using NMR spectroscopy. In this case study we describe a novel use of Gaussian process regression, which incorporates learnt Automatic Relevance Determination (ARD) parameters embedded in the covariance function of the Gaussian Processes to rank eigengenes (representaive of a clustering result) in terms of their ability to predict metabolite profiles. We combine the transcription data from the acid stress experiment with the metabolite data on the basis of this ARD ranking. Our method highlights the very important EMP bacterial pathway that is the most commonly used series of reactions for oxidizing glucose. We observe that the rapid decrease in the concentration of *glycine betaine* shows the activation of osmoregulators which might play a key role in acid adaptation.

Overall in this thesis we show how an interdisciplinary approach allows us to grow and improve methods for organizing and analysing biological data. The resulting analysis from biological data provides useful information regarding the complex interactions within bacterial biological systems. This further provides us with several hypotheses that could start a second round of the systems biology cycle of taking information about the candidate genes from proposed hypothesis

and performing biological experiments.

7.2 Future work

The work presented in this thesis shows how a Bayesian MCMC approach can handle parameter learning, inference and model selection issues. We present a general framework for GBSSM and show how it can be used to investigate datasets from different experimental conditions. The Bayesian approach also provides the possibility of incorporating prior information, based on literature or prior experimental knowledge. However this is not formally tested in this work but can be tested by including biologically informative priors in the network inference algorithm [Steele et al., 2009].

During the course of development of the GBSSM we explored an annealed importance sampling (AIS) based inference scheme. Simulated annealing explores a traceable distribution to a distribution of interest through a sequence of intermediate distributions [Granville et al., 1994]. In past it has been used to handle problems of isolating modes in the sampling process using a Markovian chain. Work by Neal [1998] shows how Markov chain transit from such annealing sequences leads to annealed importance sampling. However after implementation of AIS for SSMs we found it computationally very expensive and therefore we did not proceed further with using it. However it could provide a potentially interesting future project to efficiently implement AIS for SSMs and find its application in systems biology projects.

For any biological experiments that results in time series measurements it would be interesting to extend the SSMs with feedback loops to time varying SSMs. The models used in thesis assume stationarity i.e the parameters do not vary with time. This can be extended by introducing dynamic parameters in the model that vary with respect to time. Learning time varying SSM might enable us to provide

us with information about when the genes are expressed. This can be very useful for biologists to explore the function of a gene in a certain time frame and to invesigate time-varying transcriptional processes.

A class of linear SSMs also known as switching SSMs (SSSMs). SSSMs are a special class of time series models that iteratively divides observation into regimes with approximately linear dynamics and then estimates the parameters of these linear regimes. SSSMs are widely used in econometrics and advanced signal processing fields and we can think of finding their applications in systems biology.

The learning algorithms for SSSMs were previously studied by [Ghahramani and Hinton, 2000] and [Whiteley et al., 2010]. Ghahramani and Hinton [2000] describe the limitations of the Expectation Maximization (EM) approach and present a variational approximation that maximizes the lower bound on the log likelihood. Recently Whiteley et al. [2010] introduced discrete particle Markov chain Monte Carlo (PMCMC) methods, these are a class of MCMC algorithms which uses particle filters to build efficient proposal distributions in high dimensions.

Our approach of infering the structure of gene networks is based on the assumption that the regulation is linear. However this could not be true in cases when the interactions are highly non-linear and this might effect the accuracy of network inference. This assumption can be extended by introducing non linear state space models [Quach et al., 2007], [Noor et al., 2012].

Bibliography

B. Alberts, J. Lewis, A. Johnson, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 2002.

M. Aoki. *State Space Modeling of Time Series*. Springer-Verlag, 1990.

M. Bansal, V. Belcastro, A. A. Impiombata, and D. Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3(78):1–10, 2007.

M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21:349–356, 2005.

J. O. Berger and L. R. Pericchi. Objective Bayesian methods for model selection: introduction and comparison. *Lecture Notes-Monograph Series*, 38:135–207, 2001.

A. A. Bhagwat, J. Tan, M. Sharma, M. Kothary, S. Low, B. D. Tall, and M. Bhagwat. Functional heterogeneity of RpoS in stress tolerance of enterohemorrhagic Escherichia coli strains. *Applied and Environmental Microbiology*, 72(7):4978–4986, 2006.

D. Blackwell. Conditional expectation and unbiased sequential estimation. *Annals of Mathematical Statistics*, 18:105–110, 1947.

F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mathew, J. Gregor, N. W.

Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao. The complete genome sequence of Escherichia coli K-12. *Science*, 277:1453–1462, 2007.

P. Brazhnik, A. Fuente, and P. Mendes. Gene networks: how to put the function in genomics. *TRENDS in Biotechnology*, 20:467–472, 2002.

S. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7:434–455, 1998.

I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M. P. Cosma. A yeast synthetic network for in vivo assessment of reverse engineering and modelling approaches. *Cell*, 137: 172–181, 2009.

C. K. Carter and R. Kohn. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83:589–601, 1996.

M. P. Castanie-Cornet, T. A. Penfound, D. Smith, J. F. Elliott, and J. W. Foster. Control of acid resistance in Escherichia coli. *Journal of Bacteriology*, 181(11): 3525–3535, 1999.

S. Chib. Marginal likelihood from the Gibbs output. *The American Statistical Association*, 90(432):1313–1321, 1995.

S. Chib. Calculating posterior distributions and model estimates in Markov mixture models. *Journal of Econometrics*, 75:79–97, 1996.

W. Chu, Z. Ghahramani, F. Falciani, and D. L. Wild. Biomarker discovery with Gaussian processes in microarray gene expression data. *BMC Bioinformatics*, 21: 3385–3393, 2005.

E. J. Cooke, R. S. Savage, D. W. Kirk, R. Darkins, and D. L. Wild. Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. *BMC Bioinformatics*, 12:399–411, 2011.

M. K. Cowles and B. P. Carlins. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434): 883–904, 1996.

F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.

S. Diamant and P. Goloubinoff. Temperature-controlled activity of DnaK-DnaJ-GrpE chaperones: protein-folding arrest and recovery during and after heat shock depends on the substrate protein and the GrpE concentration. *Biochemistry*, 37 (27):9688–9694, 1998.

Y. Eguchi, T. Okada, S. Minagawa, T. Oshima, H. Mori, K. Yamamoto, A. Ishihama, and R. Utsumi. Signal transduction cascade between EvgA/EvgS and PhoP/PhoQ two-component systems of Escherichia coli. *Journal of Bacteriology*, 186(10):3006–3014, 2004.

F. Falciani. *Microarray technology through application.* Taylor & Francis, 2007.

J. W. Foster. Escherichia coli acid resistance: Tales of an amateur acidophile. *Nature*, 2(11):898–907, 2004.

N. Friedman, M. Linial, I. Nachman, and D. Peer. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

D. Gamerman and H. F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference.* Chapman and Hall Texts in Statistical Science, 2006.

C. L. Gavaghan, J. V. Li, S. T. Hadfield, S. Hole, J. K. Nicholson, I. D. Wilson, P. W. A. Howe, P. D. Stanley, and E. Holmes. Application of NMR-based metabolomics to the investigation of salt stress in maize (Zea mays). *Phytochemical Analysis*, 22(3):214–224, 2011.

A. E. Gelfand and A. F. M. Smith. Sampling based approaches to calculating marginal densities. *Journal of American Statistical Association*, 85:398–409, 1990.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472, 1992.

Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

G. Gouesbet, H. Abaibou, L. F. Wu, M. A. Mandrand-Berthelot, and Blanco C. Osmotic repression of anaerobic metabolic systems in Escherichia coli. *Journal of Bacteriology*, 175(1):214–221, 1993.

V. Granville, M. Krivanek, and J. P. Rasson. Simulated annealing: A proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):652656, 1994.

E. Guisbert, C. Herman, C. Z. Lu, and C. A. Gross. A chaperone network controls the heat shock response in Escherichia coli. *Gene and Development: Cold Spring Harbor Laboratory Press*, 18(22):2812–2821, 2004.

T. S. Gunasekera, N. L. Csonka, and O. Paliy. Genome wide transcriptional response of Escherichia coli K-12 to continuous osmotic and heat stresses. *Journal of Bacteriology*, 190(10):3712–3729, 2008.

O. Hirose, R. Yoshida, S. Imoto, R. Yamaguchi, R. Higuchi, C. Print, and S. Miyano. Statistical inference of transcriptional module based gene networks from time course gene expression profile by using state space model. *Bioinformatics*, 24(7): 932–942, 2008.

R. Hogg, J. McKean, and A. Craig. *Introduction to Mathematical Statistics*. Pearson, 2012.

D. W. Huang, B. T. Sherman, and R. A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, 4 (1):44–57, 2009a.

D. W. Huang, B. T. Sherman, and R. A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1):1–13, 2009b.

W. Huber, A. von Heydebreck, H. Sültmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18:S96–S104, 2002.

A. M. Huerta, H. Salgado, D. Thieffry, and J. Collado-Vides. RegulonDB: a database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research*, 39:55–60, 1998.

D. Husmeier, R. Dybowski, and S. Roberts. *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Springer, 2005.

R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of Affymetrix GeneChip probe level data. *Elsevier*, 31(4):1–8, 2003.

H. Jeffreys. *The Theory of Probability*. Oxford Press, 1961.

P. G. Jones, R. A. VanBogelen, and F. C. Neidhardt. Induction of proteins in response to low temperature in Escherichia coli. *Journal of Bacteriology*, 169(5): 2092–2095, 1987.

M. Kaeriyama, K. Machida, A. Kitakaze, H. Wang, Q. Lao, T. Fukamachi, H. Saito, and H. Kobayashi. OmpC and OmpF are required for growth under hyperosmotic

stress above pH 8 in Escherichia coli. *Letters in Applied Microbiology*, 42(2):195–201, 2007.

A. A. Kalaitzis and N. D. Lawrence. A simple approach to ranking differentially expressed gene expression time courses through Gaussian process regression. *BMC Bioinformatics*, 12, 2011.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

Y. Kang, K. W. Derek, Y. Qiu, P. J. Kiley, and F. R. Blattner. Genome wide expression analysis indicates that FNR of Escherichia coli K-12 regulates a large number of genes of unknown function. *Journal of Bacteriology*, 187(3):1135–1160, 2005.

R. E. Kass and A. E. Raftery. Bayes factor and model uncertainty. *Journal of the American Statistical Association*, 90:773–795, 1995.

J. Keeler. NMR and energy levels. Technical report, University of California, Irvine, 2007.

I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp. EcoCyc: a comprehensive database resource for Escherichia coli. *Nucleic Acids Research*, 39:334–337, 2005.

P. J. Kiley and H. Beinert. The role of Fe-S proteins in sensing and regulation in bacteria. *Current Opinion in Microbiology*, 6(2):181–185, 2003.

C.-J. Kim and C. R. Nelson. *State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications*. The MIT Press, 2001.

W. D. Kuczyska, S. Kedzierska, E. Matuszewska, P. Lund, A. Taylor, B. Lipiska, and E. Laskowska. The Escherichia coli small heat-shock proteins IbpA and IbpB

prevent the aggregation of endogenous proteins denatured in vivo during extreme heat shock. *Microbiology*, 154:1757–1765, 2002.

M. Kuss, T. Pfingsten, L. Csato, and C. E. Rasmusen. Approximate inference for robust Gaussian process regression. Technical report, Max Planck Institute for Biological Cybernetics, 2005.

M. Y. Kwon and S. C. Ricke. *High-Throughput Next Generation Sequencing: Methods and Applications*. Humana Press, 2011.

P. Langfelder and S. Horvath. Eigengene networks for studying relationships between co-expression modules. *BMC Systems Biology*, 54:1–17, 2007.

B. A. Lazazzera, D. M. Bates, and P. J. Kiley. The activity of the Escherichia coli transcription factor FNR is regulated by a change in oligomeric state. *Genes and Development*, 7(10):1993–2005, 1993.

H. Lethanh, P. Neubauer, and F. Hoffmann. The small heat-shock proteins IbpA and IbpB reduce the stress load of recombinant Escherichia coli and delay degradation of inclusion bodies. *Microbial Cell Factories*, 4(1):1–6, 2005.

D. J. MacKay. *Developments in Probabilistic Modelling with Neural Networks - Ensemble Learning*. Springer, 1995.

D. J. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

M. Madigan, J. Martinko, P. Dunlap, and D. Clark. *Biology of Microorganisms*. Pearson Benjamin Cummings, 2009.

A. D. Maher, J. M. Fonville, M. Coen, J. C. Lindon, C. D. Rae, and J. K. Nicholson. Statistical total correlation spectroscopy scaling for enhancement of metabolic information recovery in biological NMR spectra. *Analytical Chemistry*, 82(2): 1083–1091, 2012.

S. Mark. *Microarray Biochip Technology*. Eaton Publishing, 2000.

K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, University of California, Berkeley, 1999.

R. M. Neal. *Bayesian Learning for Neural Networks (Lecture Notes in Statistics)*. Springer, 1996.

R. M. Neal. Annealed importance sampling. Technical report, University of Toronto, 1998.

F. C. Neidhardt, J. L. Ingraham, and M. Schaechter. *Physiology of the Bacterial Cell: A Molecular Approach*. Sinauer Associates, 1990.

A. Noor, E. Serpedin, M. Nounou, and H. N. Nounou. Inferring gene regulatory networks via nonlinear state-space models and exploiting sparsity. *IEEE/ACM Trans Comput Biol Bioinform*, 9(4):1203–1211, 2012.

I. M. Ong, J. D. Glasner, and D. Page. Modelling regulatory pathways in E. coli from time series expression profiles. *Bioinformatics*, 18:241–248, 2002.

J. A. Opdyke, J. G. Kang, and G. Storz. GadY, a small-RNA regulator of acid response genes in Escherichia coli. *Journal of Bacteriology*, 186(20):6698–6705, 2004.

C. A. Penfold and D. L. Wild. How to infer gene networks from expression profiles, revisited. *Interface Focus*, 1:857–870, 2011.

B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and D. B. Florence. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19: 138–148, 2003.

Minh Quach, Nicolas Brunel, and Florence d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23:3209–3216, 2007.

C. Rangel. *Modeling Biological Responses Using Gene Expression Profiling and Linear Dynamical Statistical Models.* Claremont Graduate University, 2003.

C. Rangel, D. L. Wild, F. Falciani, Z. Ghahramani, and A. Gaiba. Modeling biological responses using gene expression profiling and linear dynamical systems. *OmniPress*, 2001.

C. Rangel, J. Angus, Z. Ghahramani, and D. L. Wild. Modeling T-cell activation using gene expression profiling and state space models. *Bioinformatics*, 20(9): 1361–1372, 2004.

C. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

T. M. Record, E. S. Courtenay, D. S. Cayley, and H. J. Guttman. Responses of E. coli to osmotic stress: large changes in amounts of cytoplasmic solutes and water. *Trends in Biochemical Sciences*, 23:143–148, 1998.

H. Richard and J. W. Foster. Escherichia coli glutamate- and arginine-dependent acid resistance systems increase internal pH and reverse transmembrane potential. *Journal of Bacteriology*, 186(18):6032–6041, 2004.

H. T. Richard and J. W. Foster. Acid resistance in Escherichia coli. *Advances in Applied Microbiology*, 52:167–186, 2003.

H. Salgado, S. Gama-Castro, A. Martinez-Antonio, E. Diaz-Peredo, F. Sanchez-Solano, M. Peralta-Gil, D. Garcia-Alonso, V. Jimenez-Jacinto, A. Santos-Zavaleta, C. Bonavides-Martinez, and J. Collado-Vides. RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. *Nucleic Acids Research Database issue*, 32:303–306, 2004.

H. Salgado, A. Santos-Zavaleta, S. Gama-Castro, M. Peralta-Gil, M. I. Penaloza-Spinola, A. Martinez-Antonio, P. D. Karp, and J. Collado-Vides. The comprehen-

sive updated regulatory network of Escherichia coli K-12. *BMC Bioinformatics*, 7:5, 2006.

M. Sato, K. Machida, E. Arikado, H. Saito, T. Kakegawa, and H. Kobayashi. Expression of outer membrane proteins in Escherichia coli growing at acid pH. *Applied and Environmental Microbiology*, 66(3):943–947, 2000.

R. Savage, K. Heller, Y. Xu, Z. Ghahramani, W. Truman, M. Grant, K. Denby, and D. Wild. R/BHC: fast Bayesian hierarchical clustering for microarray data. *BMC Bioinformatics*, 10(1):242, 2009.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

S. L. Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *American Statistical Association*, 97(457):337–351, 2002.

P. Small, D. Blankenhorn, D. Welty, E. Zinser, and J. L. Slonczewski. Acid and base resistance in Escherichia coli and Shigella flexneri: role of rpoS and growth pH. *Journal of Bacteriology*, 176(6):1729–1737, 1994.

B. J. Smith. BOA: An R package for MCMC output convergence assessment and posterior inference. *Journal of Statistical Software*, 21, 2007.

M. E. Smoot, K. Ono, J. Ruscheinski, P. L. Wang, and T. Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27 (3):431–432, 2011.

D. J. Spiegelhalter, N. G. Best, and A. V. Linde. Bayesian measures of model complexity and fit. *Statist. Soc*, 64:583–639, 2002.

S. Spiro and J. R. Guest. FNR and its role in oxygen-regulated gene expression in Escherichia coli. *FEMS Microbiology Reviews*, 6(4):399–428, 1990.

E. Steele, A. Tucker, P. A. 't Hoen, and M. J. Schuemie. Literature-based priors for gene regulatory networks. *Bioinformatics*, 25(14):1768–1774, 2009.

O. Stegle, K. Denby, W. Wild, Ghahramani. Z., and K. Borgwardt. A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *Journal of Computational Biology*, 17(3):355–367, 2010.

D. Stekel. *Microarray Bioinformatics*. Cambridge University Press, 2003.

A. Stincone, N. Daudi, S. A. Rahman, P. Antczak, I. Henderson, J. Cole, M. D. Johnson, P. Lund, and F. Falciani. A systems biology approach sheds new light on Escherichia coli acid resistance. *Nucleic Acids Research*, 39(17):1–17, 2011.

D. Stuebs, T. M. Fuchs, B. Schneider, A. Bosserhoff, and R. Gross. Identification and regulation of cold inducible factors of Bordetella bronchiseptica. *Microbiology*, 151(6):1895–1909, 2005.

Z. Szentpetery, A. Kern, K. Liliom, B. Sarkadi, A. Varadi, and E. Bakos. The role of the conserved glycines of ATP-binding cassette signature motifs of MRP1 in the communication between the substrate-binding site and the catalytic centers. *The Journal of Biological Chemistry*, 40:41670–41678, 2004.

Y. C. Tai and T. P. Speed. A multivariate empirical Bayes statistic for replicated microarray time course data. *The Annals of Statistics*, 34:2387–2412, 2006.

J. G. Thomas and F. Baneyx. ClpB and HtpG facilitate de novo protein folding in stressed Escherichia coli cells. *Molecular Biology*, 36(6):1360–1370, 2000.

R. Tibshirani. Estimating transformations for regression via additivity and variance stabilization. *Journal of the American Statistical Association*, 83(402):394–405, 1988.

B. Ventura, C. Lemerle, K. Michalodimitrakis, and L. Serrano. From in vivo to in silico biology and back. *Nature*, 443:527–533, 2006.

R. M. Viant. Improved methods for the acquisition and interpretation of NMR metabolomic data. *Biochemical and Biophysical Research Communications*, 310 (3):943–948, 2003.

R. M. Viant, E. S. Rosenblum, and R. S. Tjeerdema. NMR based metabolomics: A powerful approach for characterising the effects of environmental stressors on organism health. *Environmental Science and Technology*, 37(21):4982–4989, 2003.

R. M. Viant, J. G. Bundy, C. A. Pincetich, J. S. Ropp, and R. S. Tjeerdema. NMR-derived developmental metabolic trajectories: an approach for visualizing the toxic actions of trichloroethylene during embryogenesis. *Metabolomics*, 1(2): 149–158, 2005.

M. E. Wall, A. Rechtsteiner, and L. M. Rocha. Singular value decomposition and principal component analysis. *A Practical Approach to Microarray Data Analysis*, pages 91–109, 2003.

N. Wang, K. Yamanaka, and M. Inouye. CspI, the ninth member of the CspA family of Escherichia coli, is induced upon cold shock. *Journal of Bacteriology*, 181(5): 1603–1609, 1999.

Y. Wang, L. Wang, Y. Suna, Y. Chena, L. Zhua, L. Guoa, B. Luoa, and H. Wang. Disrupted ompC causes osmosis sensitivity of Escherichia coli in alkaline medium. *Journal of Genetics and Genomics*, 34(12):1131–1138, 2007.

A. Weber, S. A. Koegl, and K. Jung. Time-dependent proteome alterations under osmotic stress during aerobic and anaerobic growth in Escherichia coli. *Journal of Bacteriology*, 188(20):7165–7175, 2006.

N. Whiteley, C. Andrieu, and A. Doucet. Efficient bayesian inference for switching state-space models using discrete particle markov chain monte carlo methods. *arXiv preprint arXiv:1011.2437*, 2010.

F. X. Wu, W. J. Zhang, and A. J. Kusalik. Modelling gene expression from microarray expression data with state space equations. *Pacific Symposium on Biocomputing*, 581(92):581–592, 2004.

H. Xiong and Y. Choe. Structural systems identification of genetic regulatory networks. *Bioinformatics*, 24(4):553–560, 2008.

K. Yamanaka. Cold shock response in Escherichia coli. *Journal of Molecular Microbiology and Biotechnology*, 1(2):193–202, 1999.

D. E. Zak, G. E. Goney, J. S. Schwaber, and F. J. Doyle. Importance of input perturbation and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insight from an identifiability analysis of an in silico network. *Genome Research*, 13:2396–2405, 2003.